i

f

Stefan Rave, Vogelpothsweg 108, D–44227 Dortmund
Matr.-Nr. 0043401, Fachbereich Informatik, Universität Dortmund

**Danksagung**

An dieser Stelle möchte ich den Mitarbeitern und Studenten des Lehrstuhls XII, insbesondere meinen Betreuern, für die angenehme Zusammenarbeit und das jederzeit hervorragende Arbeitsklima danken.

Mein besonderer Dank gilt meinem Betreuer Birger Landwehr, der mich bei der Durchführung dieser Arbeit intensiv begleitet und unterstützt hat.

# Inhaltsverzeichnis

# Kapitel 1

# Einführung

## 1.1 Motivation

In den vergangenen Jahrzehnten hat die Mikroelektronik immense Verbreitung erfahren. Täglich kommen wir mit unzähligen digitalen Systemen in Kontakt, häufig ohne es zu ahnen. Kaum ein Gerät, das nicht in irgendeiner Weise von ihren Errungenschaften profitierte: Fotoapparat, Waschmaschine und Telefon sind nur wenige Beispiele. Schon kommen einzelne Pkw mit 100 integrierten digitalen Prozessoren daher, von denen der Bordcomputer nur das offensichtlichste ist.

Bei Schaltungen, die in der geschilderten Weise mit ihrer Umwelt interagieren, sie auswerten und kontrollieren, spricht man von *eingebetteten Systemen*. Da ihr Einsatzzweck oft ein sehr spezieller ist, kann bei ihrer Implementierung aus Gründen der Kosten, Platzbedarf und Geschwindigkeit nicht immer auf Standardkomponenten zurückgegriffen werden; ASICs[1] sind häufig die bessere Wahl. Ihre Entwicklung erfordert allerdings einen beträchtlichen Aufwand an Zeit und Geld, den es im Interesse einer schnellen Markteinführung (*Time to Market*) zu minimieren gilt.

Insbesondere sind kostenintensive Designzyklen zu vermeiden, daher müssen Entwurfsfehler möglichst frühzeitig aufgedeckt werden. Mit dem Computer läßt sich zwar das Verhalten der noch im Entwurfsstadium begriffenen Schaltung auf einzelnen Test-Eingaben (Stimuli) simulieren, aber eine derartige Simulation ist um Größenordnungen langsamer als die simulierten Vorgänge selbst; daher können nur Bruchteile der denkbaren Betriebssituationen geprüft werden. Überdies sagt ein Testlauf im Computer nichts über das Zusammenspiel der simulierten Hardware mit der späteren physikalischen Einsatzumgebung aus.

Andererseits ist es unmöglich, versuchsweise einige Vorab-Exemplare des geplanten Chips zu fertigen, denn dieser Vorgang kann mehrere Wochen in Anspruch nehmen, und die Vorbereitungen der eigentlichen Produktion verursachen hohe Kosten. Ideal wäre also eine Methode, die es gestattet, schon in frühen Entwurfsphasen und mit geringem Zusatzaufwand einen Prototyp mit der Funktionalität der spezifizierten Schaltung in Hardware zu realisieren. Ein solcher Prototyp ließe sich an seiner Stelle in der vorgesehenen Einsatzumgebung unter „Real World"-Bedingungen testen. Dieser Idealfall wird nie ganz

[1] Application Specific Integrated Circuit, anwendungsspezifischer Schaltkreis

erreicht werden können, etwa hinsichtlich Gewicht, Größe oder Leistungsaufnahme, aber je näher man ihm kommt, umso vielfältiger sind die Einsatzmöglichkeiten der Prototyping-Hardware.

Heutige mikroelektronische Schaltungen werden in VLSI-Technik[2] gefertigt, die die Integration sehr komplexer Strukturen auf kleinstem Raum gestattet. Daraus ergibt sich unmittelbar die Notwendigkeit, daß ein geeignetes Prototyping-System auch in der Lage sein muß, derart umfangreiche Strukturen in Hardware nachzubilden, ohne allerdings die hohen Kosten der eigentlichen Chip-Produktion zu verursachen. Ein anderes Merkmal ist die Geschwindigkeit: Die zuverlässigsten Aussagen über das getestete System lassen sich erst dann treffen, wenn der Hardware-Prototyp unter Echtzeitanforderungen in der vorgesehenen Einsatzumgebung betrieben werden kann.

Weil der Prototyp ein unfertiges, noch im Entwicklungsstadium befindliches System nachbildet, ist es wichtig, daß Hilfsmittel für die Fehlersuche zur Verfügung stehen. Das interne Verhalten des Prototypen sollte sich von außen nachvollziehen lassen, damit Entwurfsfehler lokalisiert werden können.

Schließlich muß der zeitliche und finanzielle Aufwand zur Realisierung des Prototypen so gering wie möglich sein, denn der Grundgedanke des Prototyping ist ja, die teure Produktion von Vorab-Mustern zu ersetzen. Je schneller ein Prototyp erstellt werden kann, desto häufiger lassen sich gründliche Vorab-Tests durchführen, und umso genauer kann das Endprodukt den Zielanforderungen entsprechen. Insbesondere ermöglichen Prototypen die Berücksichtigung von Erfahrungen der Endanwender, was einen Vorteil gegenüber Konkurrenzprodukten bedeuten kann (*Requirements Engineering*).

Die genannten Anforderungen lassen sich in hervorragender Weise mit FPGAs[3] umsetzen, auf die im folgenden eingegangen wird.

## 1.2 Einsatz von FPGAs im Rapid Prototyping

Nachdem nun die Anforderungen an Prototyping-Systeme bekannt sind, wird im folgenden Abschnitt näher auf FPGAs und ihre Eignung für das Rapid Prototyping eingegangen. In 1.2.2 schließlich wird aufgezeigt, wie eine Entwurfsumgebung für die Unterstützung von der Spezifikation bis zum Prototyp aussehen kann.

### 1.2.1 Einführung in die FPGA-Technologie

Bereits Ende der sechziger Jahre wurde das Prinzip eines konfigurierbaren Schaltkreises angedacht, doch wurde seine Realisierung erst viele Jahre später mit den großen Fortschritten auf dem Gebiet der Mikroelektronik möglich. 1985 führte der Hersteller Xilinx die FPGAs ein, Mikrochips, deren Funktionalität nicht ab Werk festgelegt ist, sondern die vom Anwender selbst festgelegt und in den Chip eingebracht wird.

[2] Very Large Scale Integration, Höchstintegration
[3] Field Programmable Gate Arrays, etwa „vor Ort programmierbare Logikgatter-Felder“

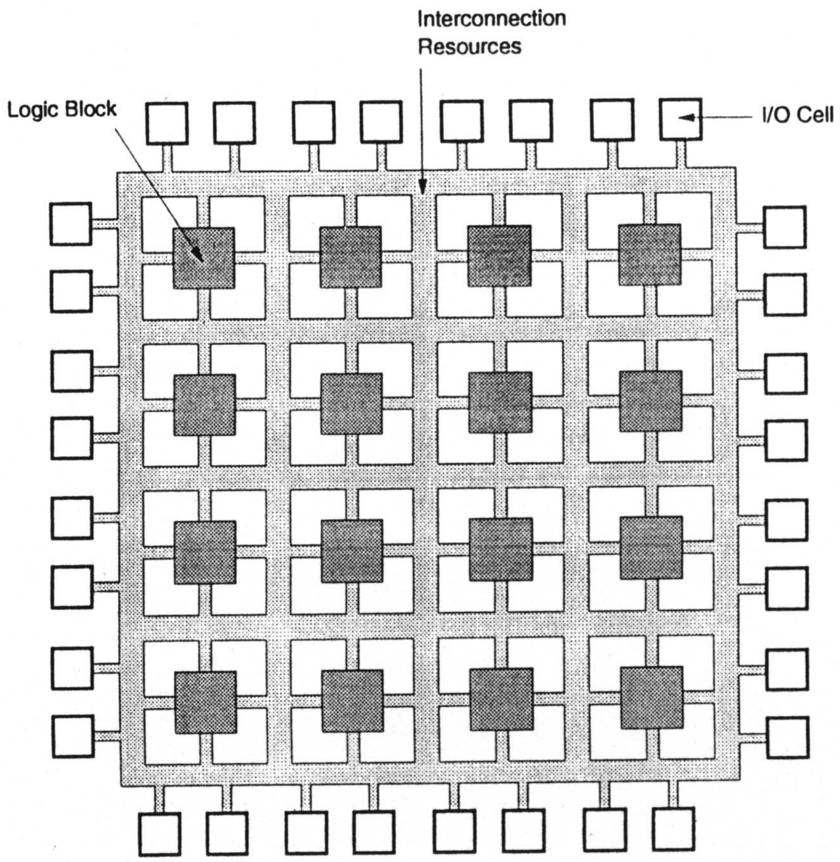


Abbildung 1.1: Konzeptioneller Aufbau eines FPGA (aus: [BFRV92])

Diese Fähigkeit verdanken FPGAs ihrem inneren Aufbau (vgl. Abbildung 1.1): Ein zweidimensionales Feld von Logik-Elementen (*Configurable Logic Blocks* oder kurz CLBs genannt) läßt sich mittels geeigneter Verbindungsressourcen flexibel verknüpfen; außenherum plazierte I/O-Zellen stellen den Kontakt mit den Anschlußpins her. Die jedes CLB umgebenden Verbindungsressourcen bestehen aus einzelnen Leitungssegmenten unterschiedlicher Länge, die mittels programmierbarer Schalter untereinander verbunden werden.

Abbildung 1.2 zeigt beispielhaft die kürzesten Leitungssegmente der im Rahmen dieser Diplomarbeit eingesetzten Xilinx XC4000-Serie. Jedes X repräsentiert einen möglichen Anschluß des CLB an ein Leitungssegment; die Segmente ihrerseits lassen sich in Schaltmatrizen flexibel miteinander verbinden.

Auch die Logikblöcke enthalten programmierbare Elemente, mit denen ihre Funktionalität vorgegeben werden kann. In Abbildung 1.3 sind links und oben die Eingänge, rechts die Ausgänge eines XC4000-CLB zu erkennen. Er besteht im wesentlichen aus *Lookup Tables* (LUT), Multiplexern und Flip-Flops. Die LUTs erlauben die beliebige logische Verknüpfung ihrer Eingänge, und die Multiplexer ermöglichen verschiedene Verdrahtungsmuster innerhalb des CLB. Die mit „Selector“ gekennzeichnete Komponente erlaubt das beliebige Verknüpfen ihrer Ein- und Ausgänge. Alle diese Elemente sind konfigurierbar.

FPGAs unterscheiden sich in der Implementierung ihrer konfigurierbaren Elemente. Im wesentlichen kommen drei Varianten zum Einsatz:

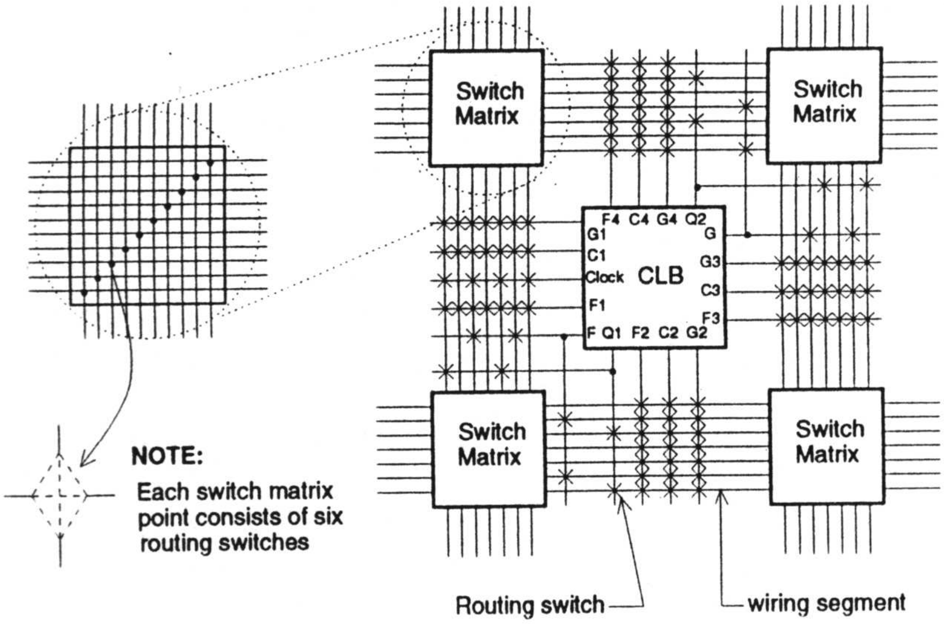


Abbildung 1.2: *Single Length Lines* der XC4000-Serie (aus: [BFRV92])

- Anti-Fuses (Hersteller Actel, Crosspoint, QuickLogic) sind spezielle Silizium-Strukturen, die im Normalzustand einen hohen Widerstand repräsentieren, durch einmaliges Anlegen einer Programmierspannung aber leitfähig gemacht werden können. Sie lassen sich platzsparend implementieren, aber nur einmal programmieren.

- EPROM[4]-basierte FPGAs (Altera, AMD, Plus) nutzen herkömmliche EPROM-Zellen zur Speicherung der Konfigurationsbits. Diese Technik verbraucht etwas mehr Chipfläche als Anti-Fuses, mit dem Vorteil der wiederholten Programmierbarkeit.

- Als dritte Variante kommen SRAM[5]-Zellen zum Einsatz (Algotronix, Concurrent, Plessey, Xilinx). Derartige FPGAs lassen sich genau wie SRAM-Speicher beliebig häufig und sehr schnell umprogrammieren; diese Technik beansprucht aber auch die meiste Chipfläche.

Die durch diese Konfigurationsressourcen gewonnene Flexibilität hat ihren Preis: Der höhere Platzverbrauch hat zur Folge, daß FPGAs nicht an die logische Kapazität von ASICs heranreichen. Als Maß für die Kapazität eines FPGA werden gemeinhin die *Gatteräquivalente* benutzt. Aktuelle FPGAs erlauben die Implementierung der Funktionalität von etwa 100.000 Gatteräquivalenten, d. h. sie können Schaltungen realisieren, die bei ausschließlicher Verwendung von NAND-Gattern mit je zwei Eingängen 100.000 solcher Elementarkomponenten erfordern würden. (Diese Angabe der Gatteräquivalente ist nur

[4]Electrically Programmable Read Only Memory, elektrisch programmierbarer Nur-Lese-Speicher

[5]Static Random Access Memory, statischer Speicher (im Gegensatz zu dynamischem Speicher, dessen Inhalt periodisch aufgefrischt werden muß)

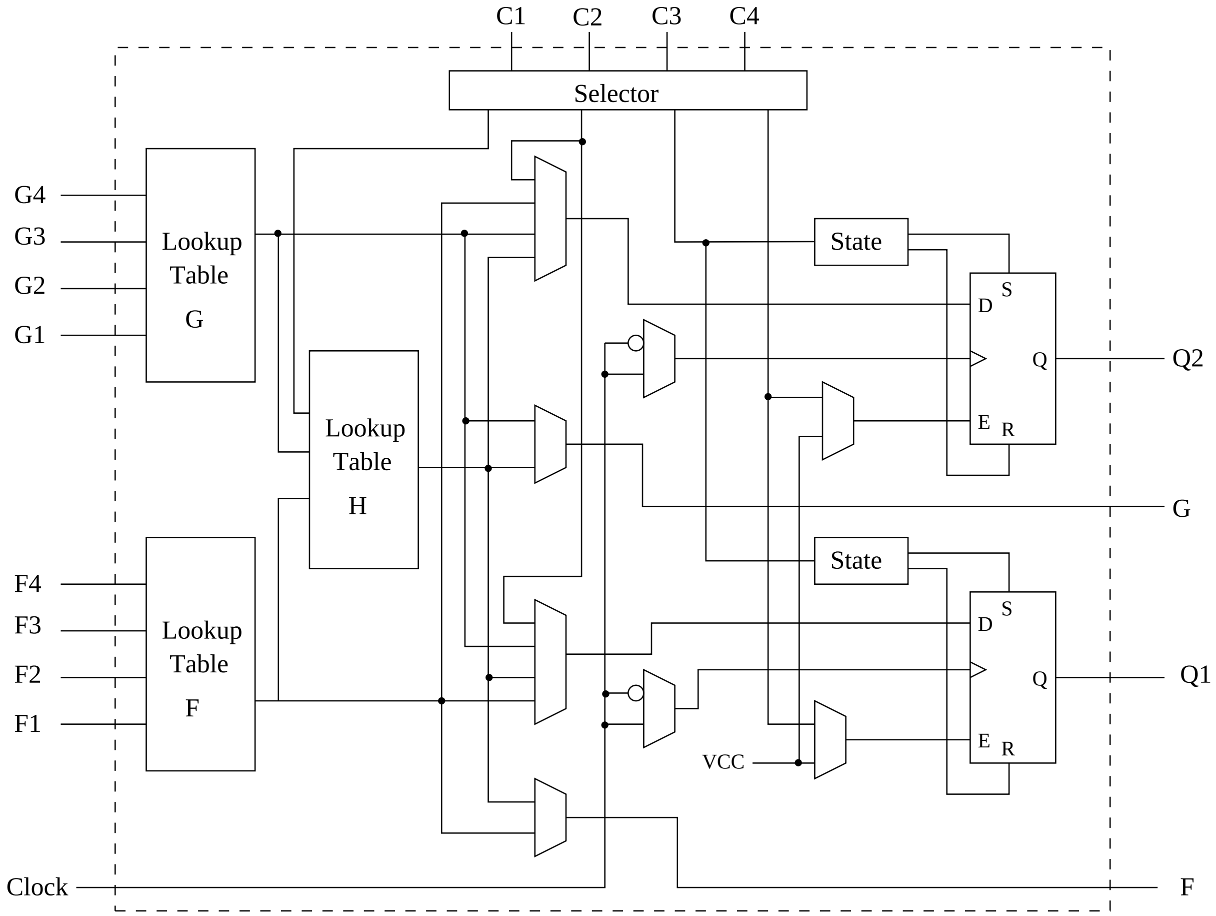


Abbildung 1.3: CLB der XC4000-Serie

ein grober Richtwert; die tatsächlich erreichbare Komplexität hängt von der jeweiligen Schaltung ab und davon, wie gut sie sich auf die Ressourcen des jeweiligen FPGA abbilden läßt.) Aktuelle ASICs hingegen werden mit weit höheren Komplexitäten bis hin zu einigen Millionen Gattern gefertigt.

Ferner stellen die Schalter, mittels derer die Leitungssegmente verknüpft werden, parasitäre Kapazitäten dar, was die Verarbeitungsgeschwindigkeit verringert. Dennoch können schnelle FPGAs — je nach implementierter Schaltung — Taktfrequenzen in der Größenordnung um 100 MHz erreichen.

Durch geeignete Programmierung des FPGA läßt sich jede beliebige Schaltung im Rahmen einer gewissen Maximalgröße realisieren. Das Erstellen einer FPGA-Konfiguration kann manuell erfolgen, was ähnlich der Assembler-Programmierung auf Kosten langer Entwicklungszeiten die effizientesten Resultate ermöglicht, oder es können Software-Werkzeuge eingesetzt werden, die den Schaltungsentwurf automatisch in CLB-gerechte Einheiten zerlegen und alle erforderlichen Daten für die Programmierung des FPGA generieren. Diese softwaregestützte Arbeitsweise ist Teil des im folgenden beschriebenen Designflusses.

### 1.2.2 Möglicher Designfluß eines Rapid-Prototyping-Systems

Mit dem Versuch, möglichst viele Arbeitsschritte zu automatisieren und vom Entwickler auf den Computer zu übertragen, geht der Trend zu immer höheren Abstraktionsebenen bei der Spezifikation von VLSI[6]-Systemen [Gaj94][Mar93][Tei97]. Während sich der Designer früher mit der unmittelbaren *Struktur* der Hardware befaßte, ermöglichen ihm moderne Entwicklungsumgebungen die Beschreibung des gewünschten System*verhaltens*. Diese Entwicklung zeigt Parallelen zur Software-Entwicklung, wo der Programmtext in intuitiven Hochsprachen vorgegeben und vom Computer in maschinentauglichen Code übersetzt (compiliert) wird.

Auch in der Hardware-Entwicklung sind Hochsprachen ein geeignetes Mittel, um das Systemverhalten zu spezifizieren. Von besonderem Nutzen ist dies im Rapid Prototyping: Die Schaltung muß nicht getrennt für die Prototyping-Plattform und das eigentliche Zielsystem entwickelt werden, sondern sie wird auf einer abstrakteren, für beide Ziele gleichermaßen geeigneten Ebene angegeben. Geeignete Software-Tools übernehmen dann in mehreren Schritten die Abbildung auf die jeweilige Hardware. Der Prototyp kann also unter Verwendung desselben Quellcodes zumindest teilautomatisch erzeugt werden.

Abbildung 1.4 zeigt beispielhaft einen denkbaren Designfluß, der einige Software-Tools kombiniert, um eine hochsprachliche Systemspezifikation in mehreren Schritten auf eine Hardware-Plattform zu transformieren. Eine solche Plattform kann aus Standardprozessoren, ASICs und FPGAs als Prototyping-Hardware bestehen. Der Schwerpunkt der Betrachtung liegt hier auf dem Teil des Prozesses, der sich mit der Prototyping-Hardware befaßt (in der Abbildung links).

Die Spezifikation des Systemverhaltens erfolgt im Beispiel in der verbreiteten Hardwarebeschreibungssprache VHDL[7]. Mittels des Codesign-Werkzeugs COOL [Nie98] wird der Entwurf auf geeignete Weise in Hardware- und Software-Komponenten partitioniert. (Dieser Ansatz des *Hardware/Software Co-Design* ist derzeit Gegenstand intensiver Forschungsbemühungen; für weitere Informationen wird auf [SW97][Buc98] verwiesen.) Die Software-Anteile werden in der Programmiersprache C ausgegeben und können mit herkömmlichen oder auch retargierbaren Compilern [Leu97] für die jeweiligen Zielprozessoren übersetzt werden.

Die Beschreibung der Hardware-Anteile liegt auf einer abstrahierenden, weil verhaltensorientierten Ebene vor. In einem nächsten Schritt muß daraus eine sogenannte RTL[8]-Netzliste synthetisiert werden, die sich ausschließlich auf reale Hardware-Konzepte wie Register und Multiplexer, Rechen- und Steuerwerke stützt; dies ist das Problem der *Mikroarchitektursynthese* [MLD92][GDWL92][Mar93]. In dem dargestellten Designflow wird sie mit dem OSCAR-System [Lan98] bewältigt.

Die so gewonnene RTL-Spezifikation macht zu diesem Zeitpunkt noch immer keine Annahmen über die konkret zum Einsatz kommende Zielarchitektur. Mittels SYNOPSYS [Syn93] kann sie in eine FPGA-orientierte Gatter-Netzliste im XNF[9]-Format überführt werden. Ist diese zu groß, um auf einem einzelnen FPGA implementiert zu werden, so läßt

[6] Very Large Scale Integration, Höchstintegration
[7] VHSIC Hardware Description Language; VHSIC: Very High Speed Integrated Circuit
[8] Register Transfer Logic
[9] Xilinx Netlist Format

sie sich mit dem Partitionierungstool COBRA [Fal98] auf mehrere miteinander gekoppelte FPGAs aufteilen.

Abschließend werden die so gewonnenen Netzlisten mit der vom Hersteller Xilinx angebotenen XACT-Software in die für die Programmierung der FPGAs erforderlichen Bitströme konvertiert.

In der Praxis werden natürlich immer wieder Re-Designs erforderlich; solche Designzyklen ließen sich im Diagramm repräsentieren durch nach oben weisende Pfeile zu früheren Entwurfsstadien. Ebenso sind die notwendigen Simulationsschritte in den Zwischenphasen nicht eingezeichnet, die üblicherweise auf allen Abstraktionsebenen durchgeführt werden.

## 1.3 Weitere Einsatzgebiete

FPGAs sind mit ihrem universellen Konzept programmierbarer Hardware natürlich für vielfältige Anwendungen geeignet. Allgemein spricht man bei Systemen, die im wesentlichen auf FPGAs basieren, von Custom Computing Machines oder kurz CCMs. Das Rapid Prototyping ist nur *ein* Einsatzgebiet solcher CCMs; daneben eignen sie sich prinzipiell auch für andere Verwendungszwecke. Das gilt uneingeschränkt auch für das hier vorgestellte FPGA-Board.

Obwohl FPGAs — wie gesehen — weder an die logische Kapazität noch an die Taktfrequenzen moderner ASICs heranreichen, können sie doch häufig höhere Gesamtleistungen erzielen als Standard-Prozessoren. In [GN96] wird die Implementierung eines Genetischen Algorithmus zur Lösung des Problems des Handlungsreisenden auf zwei verschiedenen Zielplattformen beschrieben: Die eine ist das Splash-2-System mit 4 FPGAs (zusammen ca. 40.000 Gatteräquivalente) und 11 MHz Taktfrequenz, die andere eine PA-RISC-Workstation von Hewlett Packard mit 125 MHz. Die FPGA-Variante läuft etwa viermal schneller ab, denn sie benötigt nur etwa ein Fünfzigstel der Taktzyklen der Workstation-Implementierung. Hauptursache ist die feingranulare Parallelität der Verarbeitung in den FPGAs, die sich aus der optimalen Anpassung an die jeweilige Problemstellung ergibt. Offensichtlich können Standard-Prozessoren hiermit nicht konkurrieren.

Ein weiteres Beispiel für die Tauglichkeit von FPGAs als Hochleistungsrechner für Spezial-Anwendungen ist, daß der aus der Kryptographie bekannte RSA-Code erstmalig 1990 gebrochen wurde, und zwar mit einem Verbund von 24 FPGAs des Typs XC3090. Weil derartige FPGA-basierte Spezial-Lösungen einen höheren Entwicklungsaufwand erfordern als die Software-Entwicklung für Standardprozessoren, kommen sie allerdings eher in Ausnahmefällen in Betracht.

Speziell die SRAM-basierten FPGA-Varianten gestatten die *dynamische Rekonfiguration*, d. h. die implementierte Schaltung läßt sich während des Betriebs auswechseln. Das bedeutet einen völlig neuartigen Freiheitsgrad und eröffnet sehr interessante Möglichkeiten in der Entwicklung von Elektronik, wenngleich noch nicht abzusehen ist, wie sehr solche Methoden Einzug in die Praxis halten werden.

Auf FPGAs selbst bezogen läßt sich feststellen, daß sie immer häufiger auch in Endprodukten Verwendung finden, wo sie ASICs (besonders lohnend in Kleinserien) oder Ansammlungen von Standard-Logik-ICs ersetzen. Und schließlich sind FPGAs auch ideal geeignet, um ausgelieferte elektronische Geräte vor Ort beim Kunden zu aktualisieren, also ein Hardware-Update durchzuführen. Auf diese Weise lassen sich zu spät entdeckte Fehler beseitigen oder neue Leistungsmerkmale hinzufügen.

## 1.4 Beispiele existierender FPGA-Boards

Seit Einführung der FPGAs wurden zahlreiche auf ihnen basierende CCMs entwickelt, teilweise für sehr spezielle Aufgaben, teilweise für den allgemeinen Anwendungsfall. Im folgenden werden beispielhaft einige ausgewählte, skalierbare Systeme vorgestellt, die die Eigenschaft des modularen Aufbaus gemeinsam haben, um einen Eindruck von den grundlegend verschiedenen möglichen Entwurfsansätzen und ihren Charakteristika zu vermitteln.

### 1.4.1 WEAVER

Das *WEAVER*-System [KKR94][FZI] (vgl. Abbildung 1.5) wurde mit dem Ziel der Unterstützung des Hardware/Software-Codesign entworfen, ist aber nicht auf dieses Einsatzgebiet beschränkt. Das Grundmodul (a) mit einer logischen Kapazität von ca. 100.000 Gatteräquivalenten enthält vier XC4025-FPGAs und vier Interfaces, über die sich weitere Grundmodule, Speicher-Boards oder andere Erweiterungen anschließen lassen. Durch den Einsatz von Bus-Platinen (b) können dreidimensionale Strukturen realisiert werden.

Verschiedene Merkmale dieses Grundkonzeptes erscheinen als sehr geeignet für das im Rahmen dieser Arbeit zu entwickelnde Board, doch für einen sinnvollen Einsatz des WEAVER-Systems sind verschiedene unterstützende Platinen erforderlich, in erster Linie ein Controller-Board sowie externer Speicher. Da die Interfaces des WEAVER-Systems proprietär sind, kann für derartige Hilfsmodule nicht auf bestehende, standardisierte Hardware zurückgegriffen werden. Für die hier vorliegenden Zwecke ist das von Nachteil.

### 1.4.2 ArMen

Das ArMen-Board [DFPR94][CPP$^+$94] enthält einen T800-Transputer, ein FPGA der Xilinx XC3000-Serie (ca. 7.000 Gatteräquivalente) sowie bis zu 4 MByte SRAM. Es ist mit vier Interfaces ausgestattet; eines davon befindet sich an der Rückseite der Platine und wird in eine Backplane gesteckt, so daß mehrere ArMen-Boards zu einem Bus verschaltet werden. Ein weiteres Interface dient der Verbindung mit externer Hardware wie z. B. DSP-Subsystemen. Die beiden letzten Interfaces schließlich sind an der Frontseite der Platine angebracht und werden normalerweise dafür verwendet, alle beteiligten Boards zu einem Ring zu verbinden.

Die FPGAs agieren in jedem einzelnen Board als lokaler Coprozessor, in ihrer Gesamtheit aber durch ihre Verbindung zu einem Ring als geteilter, „globaler Coprozessor“. Auf diese Weise wird ein massiv paralleler MIMD[10]-Rechner gebildet, der sich für allgemeine

[10] Multiple Instruction, Multiple Data: Gemäß der gebräuchlichen, groben Klassifikation von Parallelrechnern nach [Fly66] ein System mit mehreren Befehls- und Datenströmen.

Parallelrechner-Anwendungen sowie die Entwicklung von Echtzeitsystemen eignet. Das ArMen-System ist ein Beispiel für den feingranularen Ansatz, ein Gesamtsystem aus vielen kleinen Komponenten (Boards mit nur je einem FPGA) zusammenzusetzen.

### 1.4.3 Data Flow Functional Computer

Ein ganz anderer Ansatz wurde beim „Data Flow Functional Computer" (DFFC) [QKSZ94] verfolgt, der speziell für den Einsatz in der Echtzeit-Bildverarbeitung ausgelegt ist. Er besteht im wesentlichen aus einem Cube-Netz von 8 x 8 x 8 FPGAs, das in Form von 8 Platinen zu je 8 x 8 FPGAs organisiert ist. Dieses FPGA-Netzwerk bietet eine logische Kapazität von etwa 17 Mio. Gatteräquivalenten und wird über spezielle Controller-Hardware an einen VMEbus angeschlossen. Ergänzend kann es um Transputer-Netzwerk- sowie Speicher-Boards erweitert werden. Das DFFC-System wird benutzt, um digitale Video-Daten zu verarbeiten. In jedem Einzelschritt werden alle Iterationen des Algorithmus parallel ausgeführt, die für die Bearbeitung eines einzelnen Pixels erforderlich sind. Somit stellt der DFFC einen der wenigen Vertreter der MISD[11]-Rechner dar.

### 1.4.4 Weitere Systeme

Mittlerweile sind derart viele Boards für die unterschiedlichsten Einsatzgebiete entwickelt worden, daß eine erschöpfende Übersicht den Rahmen dieser Arbeit sprengen würde. Trotzdem sollen einige spezielle Architekturen nicht unerwähnt bleiben.

In [WH95] wird ein Computersystem beschrieben, das ein FPGA als Prozessor mit variablem Befehlssatz einsetzt, der zur Laufzeit der Software geändert werden kann. [BA97] verfolgt den Ansatz, daß sich der Datenstrom einer Anwendung aktiv durch einen Chip bewegt und die Schaltungsstruktur seinen Bedürfnissen anpaßt. Ähnliche Beispiele mit dem Schwerpunkt auf rekonfigurierbarer Hardware finden sich in [TCE+95] und [FT93].

[ML96] beschreibt die Implementierung eines stack-orientierten Prozessors mit von-Neumann-Architektur und ALU auf einem FPGA mit externem Speicher, wobei eine neuentwickelte Hardware-Beschreibungssprache zum Einsatz kommt. — Auch hier ließen sich noch viele weitere Beispiele anführen; eine umfassende Übersicht bietet [Guc99].

## 1.5 Zielsetzung dieser Arbeit

Ziel dieser Arbeit ist es, ein auf FPGAs basierendes System für den Einsatz als Prototyping-Plattform zu entwerfen und zu implementieren, das folgende Anforderungen erfüllt:

- Kapazität: Die logische Kapazität des Systems sollte ausreichend sein für einen praxistauglichen Einsatz.

[11] Multiple Instruction, Single Data

- Skalierbarkeit: Die logische Kapazität des Systems soll durch Zusammenschalten mehrerer Boards erweitert werden können, so daß die Größe implementierbarer Entwürfe nicht durch die Kapazität eines einzelnen Boards beschränkt ist. Auch sollte es möglich sein, das System um externen Speicher zu erweitern.

- Flexibilität: Getroffene Entwurfsentscheidungen sollten möglichst wenige Anwendungsaspekte a priori ausschließen. So ist es wünschenswert, daß das System sowohl autark als auch mittels einer Workstation konfiguriert werden kann. Mit der Option, daß sich das System zur Laufzeit selbständig umkonfiguriert, sollen die tiefgreifenden Möglichkeiten der dynamischen Konfigurierbarkeit von FPGAs unterstützt werden. Ferner sollte der Entwickler die Möglichkeit haben, den inneren Zustand der FPGAs zur Laufzeit einzusehen, um die Fehlersuche zu erleichtern (*Readback*). Auch ist die Unterstützung von *Shared Memory* sowie von Mikroprozessor- oder DSP-Modulen als Ersatz für ein FPGA zu erwägen.

- Standard-Bus: Unter Abwägung von Aufwand und Nutzen ist zu entscheiden, ob Kompatibilität zu einem verbreiteten Bus-Standard (z. B. PCI, VMEbus) gewährleistet werden kann; diese könnte im Idealfall die Interaktion mit einer breiten Palette bereits existierender Systeme ermöglichen.

Das System ist fertigungsreif zu entwickeln. Eine Fertigung etwa in der sogenannten *Wire Wrap*-Technik kommt aus Gründen der Betriebsicherheit und Geschwindigkeit nicht in Betracht.

In den folgenden beiden Kapiteln wird das im Rahmen dieser Arbeit implementierte Board vorgestellt. Zu jedem involvierten Teilaspekt werden im Kapitel „Entwurf“ zunächst Hintergrundinformationen vermittelt, um ein Verständnis für die Problemstellungen zu wecken; deren technische Umsetzung ist Thema des Kapitels „Realisierung“.

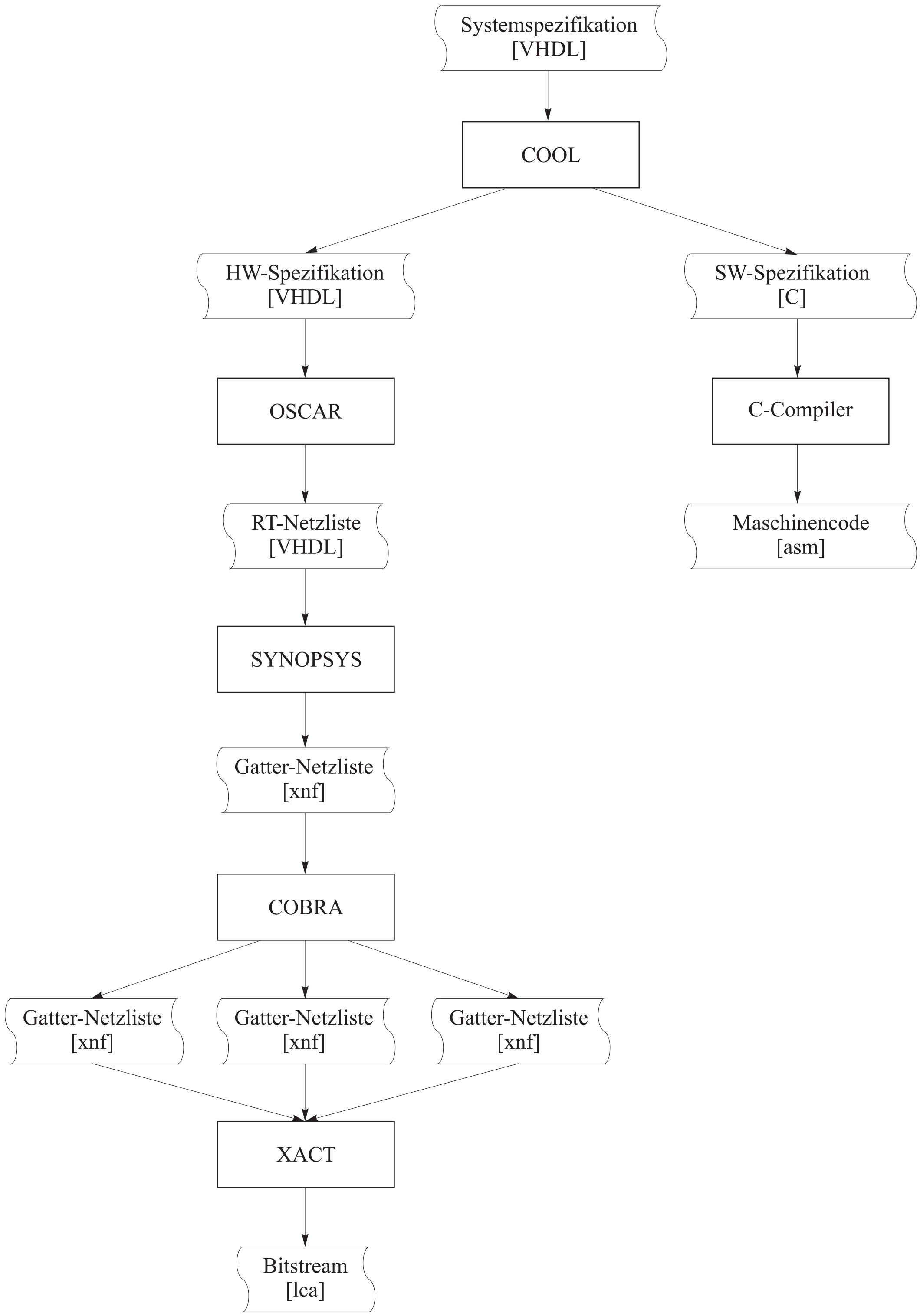


Abbildung 1.4: Möglicher Entwurfsprozeß

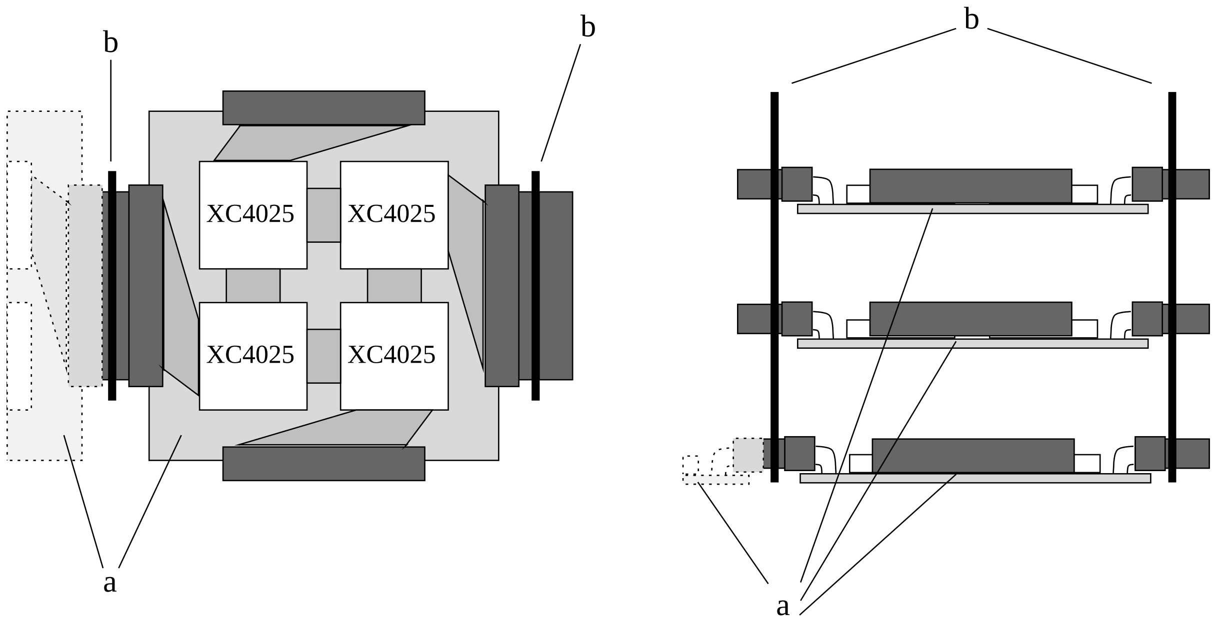


Abbildung 1.5: Das WEAVER-System

# Kapitel 2

# Entwurfsaspekte

Dieses Kapitel liefert Hintergrundinformationen zu verschiedenen Fragestellungen, die sich beim Entwurf des Systems ergeben, um ein Verständnis für die in Kapitel 3 dargestellten technischen Umsetzungen dieser Aspekte vorzubereiten.

## 2.1 Topologie

Bei der Betrachtung eines aus dem Verbund mehrerer gleichartiger Boards bestehenden Gesamtsystems stellt sich die Frage nach der Board-übergreifenden, also globalen Topologie genau wie die Frage nach der internen, also der Topologie jedes einzelnen Boards. Auf beide Aspekte wird im folgenden näher eingegangen.

### 2.1.1 Board-übergreifende Topologie

Der Prämisse folgend, daß möglichst kein Anwendungsfall von vornherein ausgeschlossen werden soll, ergibt sich die Forderung nach flexibler externer Verknüpfbarkeit jedes einzelnen Boards, um die Realisierung verschiedenster Netz-Topologien zu erlauben. Beispiele für solche Topologien sind in Abbildung 2.1 dargestellt; jeder Knoten entspricht in diesem Kontext einem FPGA-Board.

Die Variante der vollständigen Vermaschung erfordert bei $N$ beteiligten Boards offensichtlich $N - 1$ Interfaces je Board und kommt daher nur für Systeme geringer Größe in Betracht, denn die Anzahl der Interface-Pins je Board ist natürlich schon allein aufgrund geometrischer Gegebenheiten beschränkt. Relevanter sind hier Netze, bei denen jeder Knoten eine begrenzte Anzahl von Nachbarn hat. Häufig kommen Baum- oder Gitterstrukturen zum Einsatz; Ring- und Cube-Netze sind ein- bzw. dreidimensionale Gitterstrukturen. Die Bus-Topologie, bei der sich mehrere Teilnehmer ein gemeinsames Transportmedium teilen, ist wegen ihrer Einfachheit und Flexibilität von besonderer praktischer Relevanz. Sie bietet die Möglichkeit eines stabilen Aufbaus des Gesamtsystems in einem kompakten Gehäuse (*Rack*) mit Steckplätzen für die Einzelkomponenten. Diverse Bus-Standards sind definiert, zu denen eine breite Palette kompatibler Hardware existiert. Diese Gründe sprechen dafür, besonderes Augenmerk auf den Bus-Betrieb zu legen, doch soll die Unterstützung anderer, allgemeiner Gesamtsystem-Topologien nicht vernachlässigt werden.

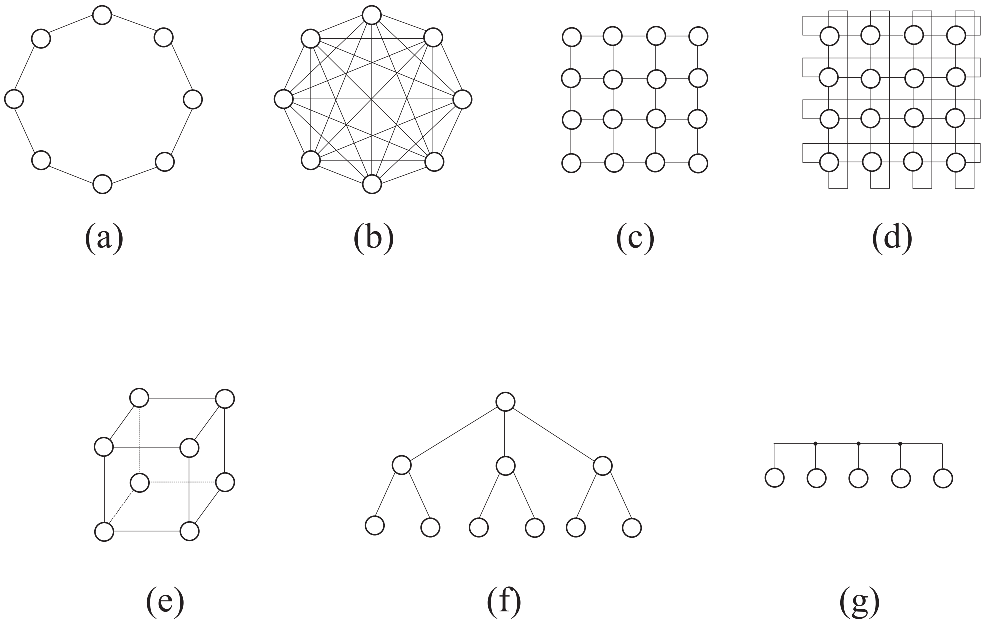


Abbildung 2.1: Verschiedene Netztopologien: (a) Ring, (b) vollständige Vermaschung, (c) Gitter, (d) Torus, (e) Cube, (f) Baum, (g) Bus

In diesem Kontext erscheint der Hinweis angebracht, daß FPGAs nicht nur als Prozessor-, sondern auch als Verknüpfungselemente eingesetzt werden können, denn ihre freie interne Programmierbarkeit erlaubt natürlich insbesondere das bloße Verbinden verschiedener Anschlüsse. Es ist denkbar, Teile eines FPGA als Rechenwerk einzusetzen, während andere als aktive, variable Koppelelemente der nicht vom Rechenwerk belegten I/O-Pins fungieren. Somit lassen sich auf ein FPGA oder auch auf einen Verbund mehrerer FPGAs direkte Netze mit rein statischen Verbindungen gleichermaßen wie indirekte Netze mit zwischengeschalteten Vermittlungselementen abbilden. FPGAs können also nicht nur als Endpunkte einer Verbindung dienen, sondern auch als Zwischenpunkte, die dynamisch variierende Verbindungen herstellen. Diese Tatsache bedeutet einen erheblichen Zugewinn an Flexibilität bei der Zusammenschaltung mehrerer Boards. (Für umfassende Informationen zu direkten und indirekten Verbindungsnetzen vgl. z. B. [SJ96]).

### 2.1.2 Board-interne Topologie

Um mehrere Boards zu Strukturen wie den oben beschriebenen verbinden zu können, muß jedes einzelne selbstverständlich mehrere Schnittstellen nach außen zur Verfügung stellen — die offensichtlichste Anforderung an die Board-interne Topologie. Ein Großteil der praxisrelevanten Netze (insbesondere Gitter- und Torus-Netze) läßt sich mit vier Schnittstellen je Board unmittelbar realisieren; mehr Interfaces versprechen nur bei sehr großen, drei- oder mehrdimensionalen Netzen einen gewissen Gegenwert für den Mehraufwand an Platz und I/O-Ressourcen. Unter Zuhilfenahme von Bus-Platinen lassen sich dreidimensionale Strukturen im übrigen auch mit nur vier Schnittstellen je Board realisieren (vgl. Abbildung 1.5). Eine Geometrie ähnlich der des in Abschnitt 1.4.1 vorgestellten *WEAVER*-Systems mit vier Interfaces erscheint bei den gegebenen Anforderungen also am sinnvollsten. Auch zu der weiteren Konsequenz, das Board mit genau vier FPGAs auszustatten und jedem davon exklusiven Zugriff auf eine eigene Schnittstelle zu geben, gibt es keine sinnvolle Alternative. Alle denkbaren Alternativen machen das Grundkonzept unnötig kompliziert, bringen aber keine erkennbaren Vorteile für den allgemeinen Anwendungsfall und lassen sich im übrigen mit gewissen Einschränkungen auch bei der 1:1-Verknüpfung von FPGAs und Interfaces erzielen durch Maßnahmen wie externe Bus-Boards oder den Einsatz einzelner FPGAs als Vermittlungsknoten.

Innerhalb des Boards müssen die FPGAs in einer Weise verbunden werden, die den verschiedenen denkbaren externen Topologien nicht im Wege steht. Wie auch bei der Board-übergreifenden Topologie bietet natürlich die vollständige Vermaschung, also die Verbindung jedes FPGAs mit allen anderen, die maximale interne Konnektivität, doch läßt sich mit spärlicheren Strukturen unter Umständen ein größerer Nutzen aus dem begrenzten Vorrat an Anschluß-Pins ziehen. Die Bus-Topologie kann auch Board-intern ihre Vorzüge ausspielen, nämlich die Verbindung aller Teilnehmer untereinander mit minimalem Aufwand an I/O-Pins. (Für eine 1-Bit-Verbindung einer Komponente mit $N$ anderen werden bei separaten Leitungen offensichtlich $N$ Pins je Komponente benötigt, während beim Bus nur 1 Pin je Komponente anfällt.) Prinzipiell wäre eine rein busbasierte Kopplung aller FPGAs innerhalb des Boards natürlich auch geeignet, um jede beliebige externe Topologie zu unterstützen, denn es kann ja jeder Bus-Teilnehmer mit jedem anderen kommunizieren. Einzelne FPGAs aber stellen weniger autarke Einheiten dar als ganze FPGA-Boards, so daß ein höherer Kommunikationsbedarf untereinander zu erwarten ist. Zumindest benachbarte FPGAs sollten daher unmittelbar miteinander kommunizieren können, ohne den Bus in Anspruch nehmen und so die übrigen FPGAs gegebenenfalls blockieren zu müssen.

Ein weiterer Aspekt der Board-internen Topologie ist die Einbindung von Speicher in die Kommunikationsstrukturen. Hierauf wird in Abschnitt 2.3 näher eingegangen.

## 2.2 Logische Kapazität und Granularität

Bei der Frage nach dem sinnvollsten Wert für die logische Kapazität des Boards sind verschiedene, teilweise konkurrierende Einflüsse zu berücksichtigen. Von fundamentaler Bedeutung ist die Tatsache, daß das System skalierbar sein soll. Damit wird die Frage nach der Kapazität des einzelnen Boards zu einer Frage nach der Granularität des Gesamtsystems: Theoretisch lassen sich durch Zusammenschalten vieler „kleiner“ Boards große Gesamt-Kapazitäten erzielen, während „große“ Boards die Gefahr in sich bergen, daß in den meisten Anwendungsfällen ein wesentlicher Teil ihrer Ressourcen brachliegt. So gesehen erscheint allein eine kleine logische Kapazität des einzelnen Boards erstrebenswert, doch sprechen andere Gründe dagegen: Steckvorrichtungen, mit denen einzelne Boards verbunden werden müssen, stellen für elektrische Signale ungünstigere Leiter dar als es die Leiterbahnen innerhalb einer Platine sind. Zwischen benachbarten Pins von Steckern kommt es zu Übersprech-Effekten, d. h. die zugehörigen Signale werden geringfügig miteinander vermischt, und außerdem begrenzen die jeder Verbindungsvorrichtung inhärente parasitäre Kapazität und Induktivität die maximal erzielbare Taktfrequenz. Je mehr Interfaces ein Signal passieren muß, umso langsamer muß der Takt des Gesamtsystems ausfallen.

Abgesehen davon sprechen auch offensichtlichere Gründe gegen Boards mit geringer Kapazität, etwa die Tatsache, daß der mechanische Aufbau im Falle vieler kleiner Einzelboards unzuverlässiger ist und mehr Raum beansprucht als ein aus wenigen leistungsfähigeren Platinen bestehendes System. Besonders in Topologien mit ausgewiesenen zentralen Bestandteilen schließlich werden eben diese leicht zum Flaschenhals, wenn das Gesamtsystem übermäßig groß wird. Wie in Abschnitt 2.1 gesehen, trifft das auf Bus-Architekturen zu, die wiederum im Rahmen dieses Projektes sinnvoll erscheinen.

Aufgrund der Erfahrungen mit verschiedenen Projekten [Pro94] [Pro96] [Pro97] erscheint eine Kapazität von etwa 100.000 Gatteräquivalenten je Board als angemessen dimensioniert und bei Verschaltung mehrerer solcher Boards zu einem Gesamtverbund geeignet, hinreichend große Entwürfe zu implementieren. Beispielsweise zeigte sich in [Pro94], daß ein manuell auf geringen Platzverbrauch optimierter effizienter 16-Bit-Multiplizierer, auf Xilinx-FPGAs gemappt, 367 CLBs beansprucht, was grob geschätzt 10.000 Gatteräquivalenten entspricht. Die erwähnten 100.000 Gatter sollten also für praxistaugliche Rechenwerke zuzüglich einiger Steuerungslogik ausreichen, und auch größere Systeme lassen sich natürlich durch Verbinden mehrerer Boards realisieren.

## 2.3 Speicher

Bisher wurden Entwurfsaspekte betrachtet, die mit den FPGAs selbst in Zusammenhang stehen. Eine ähnlich zentrale Bedeutung hat die Frage nach dem Einsatz von Speicher, und zwar nach Konfigurations- wie Arbeitsspeicher. Dies wird in den folgenden beiden Abschnitten näher beleuchtet.

### 2.3.1 Konfigurationsspeicher

Die SRAM-Zellen der eingesetzten FPGAs müssen nach jedem neuerlichen Anlegen der Versorgungsspannung zunächst mit den Konfigurationsdaten programmiert werden. Es ist möglich, diese den FPGAs von außen über ein serielles Interface zuzuführen. Will man aber ein über längere Zeit unverändertes Design einsetzen, so ist es unnötig umständlich, das Board nach jedem Einschalten mittes eines Wirtsrechners zu programmieren; daher ist bei den hier Verwendung findenden Xilinx-FPGAs ein sogenannter Master-Modus vorgesehen, in dem die FPGAs selbständig und autark einen lokal angeschlossenen, sinnvollerweise nichtflüchtigen Speicher Byte-parallel auslesen und sich so konfigurieren. In dieser Betriebsart genügt es, die Spannung einzuschalten, und der Rest geschieht von allein; der Nachteil der flüchtigen SRAM-basierten FPGAs wird so relativiert. Hierfür benötigt nur ein einzelnes FPGA Zugriff auf den Konfigurationsspeicher; es konfiguriert die übrigen FPGAs mit (vgl. Abschnitt 3.7.)

Wäre Konfigurationsspeicher auf der Platine unter dem genannten Gesichtspunkt notfalls verzichtbar, so wird er spätestens für das Leistungsmerkmal der *dynamischen Rekonfiguration* zum Muß (vgl. Abschnitt 2.5). Die Implementierung eines 8 Bit breiten, nichtflüchtigen Konfigurationsspeichers ist daher fest vorgesehen. Aufgrund des hiervon sehr verschiedenen technischen Anforderungsprofils ist die Frage nach eventuellem On-Board-Arbeitsspeicher aber getrennt zu behandeln; das geschieht im folgenden Abschnitt.

### 2.3.2 Arbeitsspeicher

Die FPGAs der XC4000-Reihe können mit ihren CLBs in geringem Umfang RAM bereitstellen, etwa für Registerbänke, doch bewegt sich dieses Feature in Größenordnungen von wenigen Kilobyte je FPGA; je mehr es in Anspruch genommen wird, umso weniger CLBs stehen für die eigentliche Logik zur Verfügung. FPGA-internes RAM ist also als sehr „teuer“ zu betrachten und sollte nur in möglichst geringem Umfang eingesetzt werden. Zusätzlicher Arbeitsspeicher ist für die meisten Anwendungen unverzichtbar und in Form von dedizierten Speicherchips in ganz anderen Größenordnungen erhältlich: SRAM-ICs sind durchaus mit Kapazitäten von 4 MBit verfügbar. (Der Einsatz von DRAM- oder SDRAM-Produkten scheidet wegen ihrer komplizierten Zugriffsmethoden aus, vgl. z. B. [Hit98].)

Anhand der Beispiele aus Abschnitt 1.4 ist ersichtlich, daß hinsichtlich der Einbindung von Arbeitsspeicher sehr verschiedene Ansätze praktikabel sind, von lokalem On-Board-Speicher für jedes einzelne FPGA bis hin zum alleinigen Einsatz von externem Speicher. Diese letztgenannte Option erfordert mehrere Interfaces je Board, damit neben externem Speicher auch weitere Hardware angeschlossen werden kann; da für das hier zu entwerfende System ohnehin mehrere Interfaces vorgesehen sind, kommt also das ganze Spektrum von Möglichkeiten in Betracht. Da aber ohne Arbeitsspeicher nur in den wenigsten Fällen ein sinnvoller Einsatz der FPGAs möglich ist, fiel die Entscheidung zugunsten von On-Board-Speicher: So bleiben alle Interfaces für andere Zwecke verwendbar, und der Aufwand (insbesondere der Platzbedarf) erscheint mehr als vertretbar.

Die Zugriffsgeschwindigkeit auf den Arbeitsspeicher ist in den meisten Fällen kritisch, gerade bei Prozessorarchitekturen, die in jedem Taktzyklus zum Holen des Befehlswortes auf den Speicher zugreifen müssen; in üblichen Mikroprozessorsystemen etwa wird mittels besonders schneller Zwischenspeicher (*Cache*) versucht, dem Prozessor in jedem einzelnen Takt einen Transfer zu erlauben. Aus diesen Gründen sollte möglichst jedes einzelne FPGA ungeteilten Zugriff auf eigenen Arbeitsspeicher haben, eine Einbindung des Speichers in eine größere Bus-Struktur kommt also nicht in Betracht. Andererseits dürfen nicht zu viele FPGA-Pins dem bloßen Speicherzugriff geopfert werden. Je Speicher-Interface sind grob geschätzt 20 Adreß- und mindestens 16, besser 32 Datenleitungen vorzusehen, zuzüglich einiger Kontrollsignale. In Abschnitt 3.3.3 wird der Lösungsweg vorgestellt, der im Rahmen dieser Arbeit eingeschlagen wurde.

## 2.4 Anbindung an Standardbus

Der Nutzen des FPGA-Boards wäre bei Kompatibilität zu einem etablierten Bus-Standard größer, weil unmittelbar auf bereits existierende Hardware zugegriffen werden könnte, beispielsweise um den Arbeitsspeicher mit einer externen Platine zu vergrößern, oder um mit anderen Prozessor-Boards zu kommunizieren. Abgesehen davon könnte von der im täglichen Einsatz bewährten Architektur eines etablierten Bus-Systems profitiert werden. Eine Standardbus-kompatible Auslegung der Interfaces ist daher grundsätzlich einer proprietären Lösung vorzuziehen, allerdings müssen auch hier Vor- und Nachteile abgewogen werden.

Aus den handelsüblichen PCs sind die Bus-Standards ISA (mit der leistungsfähigeren Variante EISA) sowie PCI bekannt. Es handelt sich in beiden Fällen um Systeme für Steckkarten, d. h. Platinen, die mit einer seitlichen Kante in Einsteckplätze (*Slots*) eingeführt werden. Dieses Merkmal macht kompatible Karten ausschließlich für den Einsatz im jeweiligen Bus-System geeignet; das flexible Verbinden mehrerer Boards etwa zu einer Matrix-Topologie ist unmöglich. Auch der in PCs eingesetzte SCSI-Bus kommt hier nicht in Betracht. Zwar wäre seine Stecker-orientierte Auslegung prinzipiell geeigneter als die Einsteckkarten des ISA- und des PCI-Busses, aber der SCSI-Bus ist für andere Zwecke gedacht: Existierende Hardware besteht praktisch ausschließlich aus Peripheriegeräten wie Festplatten, Scannern und dergleichen, während in unserem Kontext wünschenswerte DSP- oder Speicherboards und ähnliche Komponenten kaum je für den SCSI-Bus entwickelt werden.

Geeignetere Bus-Standards sind Multibus II (IEEE 1296), Nubus (Texas Instruments) und VMEbus (IEEE 1014-1987); sie alle verwenden Stecker und Buchsen nach DIN 41612, die für die hier geforderte flexible Skalierbarkeit mehrerer Boards zu einem großen Verbund sehr geeignet sind. Im Kontext des hier vorgestellten FPGA-Systems empfiehlt sich besonders der VMEbus, bzw. die in ANSI/VITA 1-1994 spezifizierte Erweiterung VME64 [VME95]. Im Gegensatz zu Nubus und Multibus II arbeitet er asynchron, und außerdem ist er am flexibelsten einsetzbar und für sehr allgemein gehaltene Zwecke ausgelegt; im folgenden wird näher auf die sich daraus ergebenden Vorteile eingegangen. Es existiert eine breitgefächerte Palette VME-kompatibler Hardware-Komponenten aus allen hier relevanten Einsatzbereichen. Abbildung 2.2 zeigt schematisch eine denkbare Konfiguration eines VMEbus-Systems. (Auch der unerreichte Adreßraum von 64 Bit spricht für den VME64-Standard, doch dürfte dies nur in Spezialfällen zum Tragen kommen.)

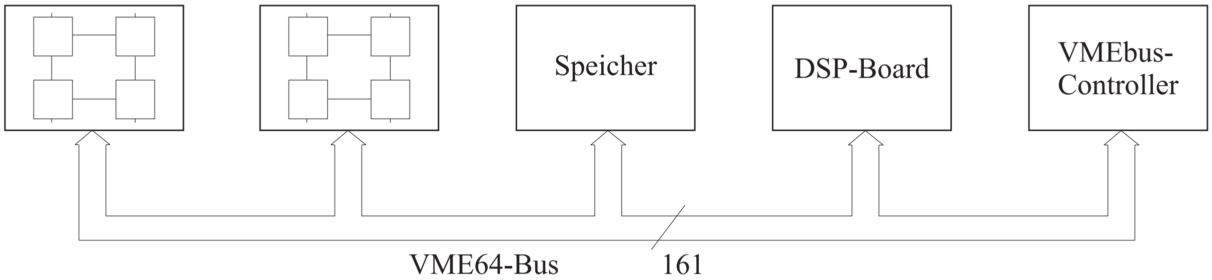


Abbildung 2.2: Beispiel eines VMEbus-Systems (schematisch)

Die asynchrone Arbeitsweise des VME-Busses hat zur Folge, daß er sich der Geschwindigkeit der angeschlossenen Komponenten anpaßt; Transfers zwischen schnellen Komponenten werden nicht durch eine fest vorgegebene Systemtaktfrequenz ausgebremst, andererseits aber können auch langsame Komponenten eingesetzt werden. Diese Eigenschaft kommt dem FPGA-Board entgegen, denn je nach eingebrachter Schaltung können die FPGAs ganz unterschiedliche maximale Verarbeitungsgeschwindigkeiten erreichen. Für den Entwickler besteht keine Notwendigkeit, eine durch das Bus-System vorgeschriebene Taktfrequenz einzuhalten.

Der VME64-Standard erlaubt verschiedene Platinenformate, von denen das *6U* oder *Doppeleuropa* genannte besonders geeignet erscheint (vgl. Abbildung 2.3): Seine Abmessungen von 160 x 233 mm entsprechen in etwa dem Platzbedarf von vier FPGAs sowie einigen

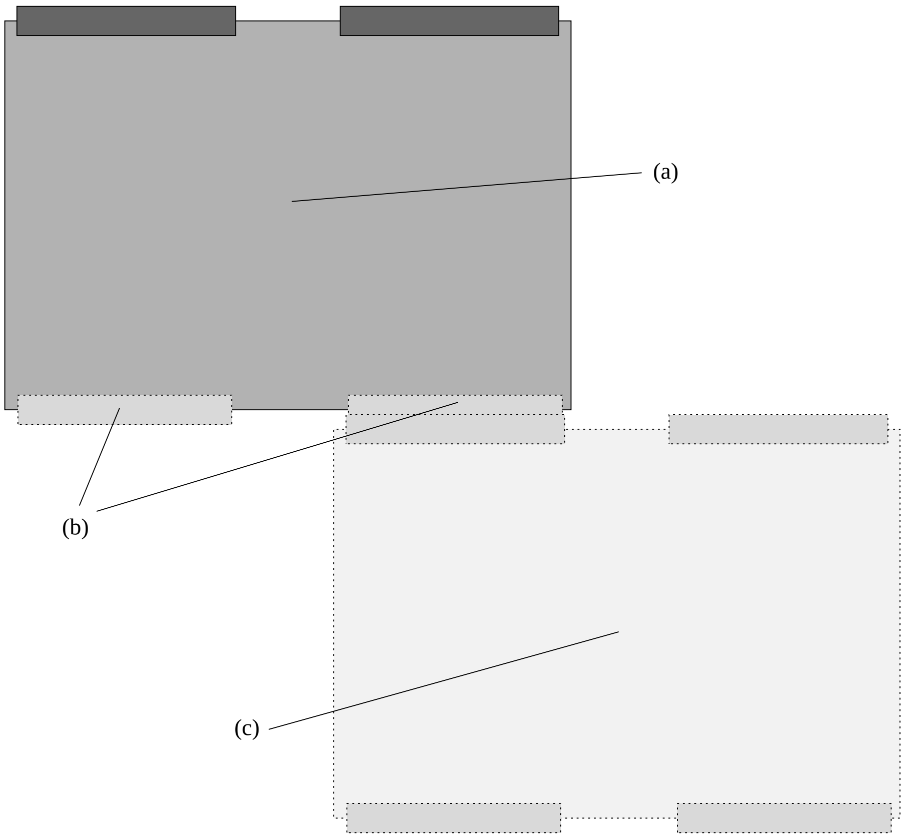


Abbildung 2.3: VME64-Platine (a) mit zusätzlichen Buchsen (b) im Verbund mit einer identischen Platine (c)

Speicher-ICs und sonstigen Bauteilen, die für das hier zu entwerfende System erforderlich sind. Darüberhinaus sind 6U-Platinen mit je zwei 96poligen Steckern (Messerleisten) ausgestattet[1]; werden zusätzlich zwei entsprechende Buchsen (Federleisten) auf der gegenüberliegenden Seite der Platine angebracht, so stehen vier Interfaces mit je 96 Pins zur Verfügung.[2] Das entspricht gut den hier vorliegenden Anforderungen an die externe Konnektivität, und es ermöglicht den Betrieb des Boards sowohl im VMEbus als auch im größeren Verbund mit weiteren FPGA-Boards.

Die Flexibilität des VME-Systems zeigt sich besonders deutlich in der Palette erlaubter Bustransfers: Je nach Anforderungen kann man zwischen einer Datenbreite von 8, 16, 24, 32 oder 64 Bit wählen; bei 64 Bit lassen sich Übertragungsraten bis 80 MByte/s erzielen. Für den Betrieb von VME-Platinen ist nur einer der beiden Stecker notwendig, der andere ist für breite Bustransfers optional zusätzlich benutzbar.

Genauere Informationen über den VME-Standard finden sich in [VME95][Pet97]. In Abschnitt 3.4.1 wird auf die technischen Maßnahmen eingegangen, die im Zusammenhang

[1]Optional sind 160polige Stecker erlaubt, doch so viele Interface-Pins werden nicht benötigt. Überdies sind 160polige, abgewinkelte Buchsen nicht erhältlich.

[2]Laut VMEbus-Standard ist an der Kopfseite der Platine eine Frontplatte mit Griffen zum Herausziehen der Platine aus dem Rack vorgesehen. Diese Frontplatte muß bei dem geschilderten Vorgehen entfallen, was der Funktionalität der Platine keinen Abbruch tut. Vorrichtungen als Griff-Ersatz lassen sich auch an den ohnehin erforderlichen Schrauben der Buchsen anbringen.

mit der VME64-Kompatibilität getroffen wurden, doch zunächst werden weitere Entwurfsaspekte behandelt.

## 2.5 Konfigurationsvorgang

Die bereits im Zusammenhang mit dem Konfigurationsspeicher erwähnte Programmierung der FPGAs läßt sich passiv (im *Slave*-Modus) über ein serielles Interface von einer Workstation aus durchführen. Ein solcher Interface-Anschluß ist für jedes einzelne FPGA vorgesehen, aber auch das Feature des *Daisy Chaining*[3] mehrerer FPGAs soll unterstützt werden. Damit das Board autark betrieben werden kann, ist auch die Unterstützung des *Master*-Modus vorgesehen. Diese Maßnahmen entsprechen dem üblichen Vorgehensweise [Alf97].

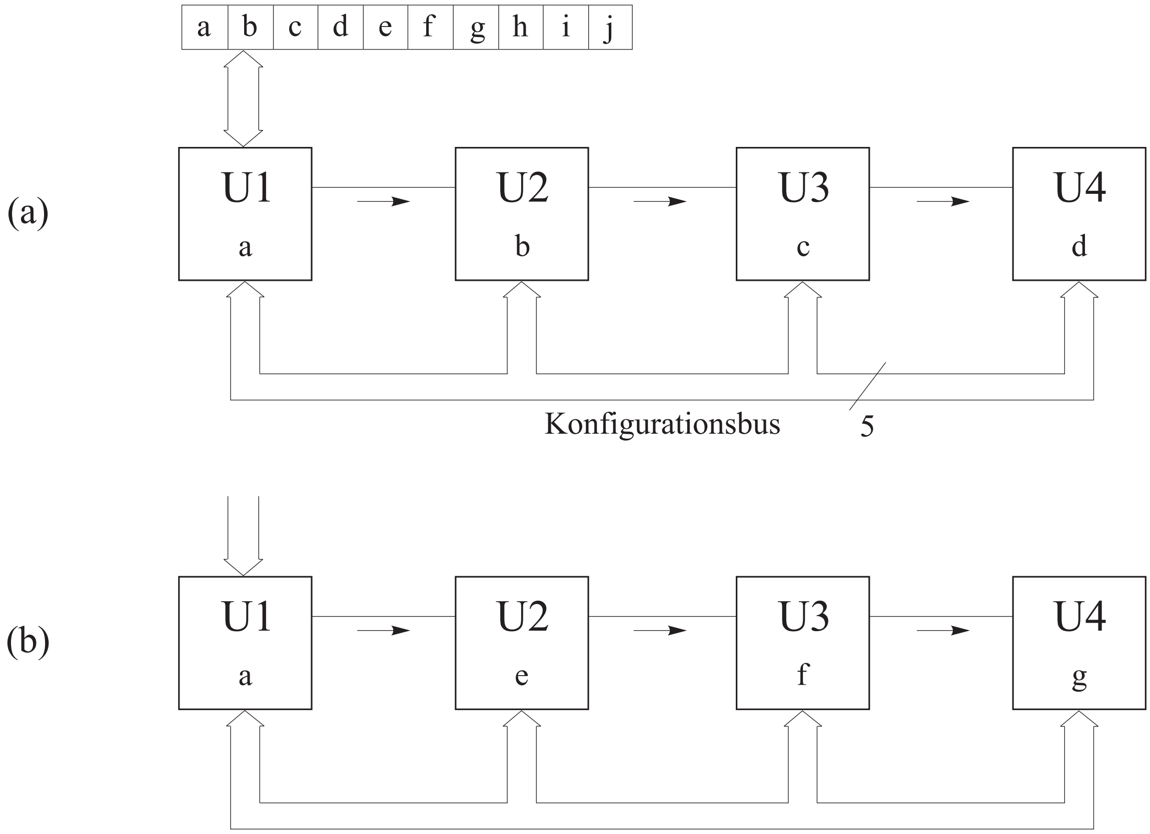


Abbildung 2.4: Dynamische Rekonfiguration von *U2* bis *U4* durch *U1*

Abgesehen davon ist für das hier behandelte Board eine Besonderheit vorgesehen, die nur in wenigen bestehenden Systemen unterstützt wird, nämlich die *dynamische Rekonfiguration*. Es soll dem Benutzer möglich sein, einige FPGAs zur Laufzeit je nach Bedarf umzuprogrammieren, so daß sie zu verschiedenen Zeitpunkten unterschiedliche Hardware implementieren. Abbildung 2.4 (a) zeigt hierzu schematisch den Zustand der vier FPGAs

[3]Daisy Chain: Gänseblümchenkette; in der Elektronik verwendeter Begriff für ein Signal, das sequentiell durch mehrere Abnehmer geleitet wird

eines Boards, nachdem *U1* im Master-Modus sich selbst und die nachfolgenden FPGAs mit den im Konfigurationsspeicher abgelegten Daten programmiert hat. Zu einem späteren Zeitpunkt kann *U1* die übrigen FPGAs nach Belieben umprogrammieren, indem es alternative Konfigurationsdaten aus dem Speicher liest. So kann ein Zustand wie in Teil (b) der Abbildung hergestellt werden. (Die Programmierung von *U1* selbst bleibt natürlich konstant.)

Alle angesprochenen Konfigurationsvarianten werden in Abschnitt 3.7 hinsichtlich ihrer technischen Umsetzung behandelt.

## 2.6 Unterstützung bei der Fehlersuche

Als letzter Entwurfsaspekt wird an dieser Stelle das Debugging betrachtet, also Maßnahmen zur Unterstützung der Fehlersuche. Diese ist in Hardware-Systemen ist ungleich aufwendiger als in reinen Software-Systemen: Im Falle der Software ist es möglich, die Ausführung mittels *Breakpoints* in definierbaren Situationen zu unterbrechen und alle relevanten programminternen Variablen zu untersuchen, bei Bedarf auch zu modifizieren und die Ausführung fortzusetzen. Das Innere eines laufenden Hardware-Systems dagegen ist normalerweise unzugänglich. Man kann allenfalls versuchen, das Systemverhalten durch Beobachtung der von außen zugänglichen Anschlußpins zu verfolgen und so Rückschlüsse auf innere Vorgänge zu ziehen, aber der Zustand eines internen Flip-Flops innerhalb eines IC ist nicht unmittelbar einzusehen. Bedenkt man, daß übliche FPGAs Zehntausende von Flip-Flops und Register-Bits (etwa in der Lookup Table) beinhalten, so leuchtet ein, daß die Suche nach der Ursache eines Fehlverhaltens ohne weitere Maßnahmen extrem aufwendig wäre.

Gerade im Kontext eines Prototypen-Systems aber sind weitergehende *Debugging*-Hilfsmittel äußerst wünschenswert, denn das System soll ja gerade dazu dienen, Schwächen und Fehler des getesteten Entwurfs in einer sehr frühen Entwicklungsphase aufzudecken.

### 2.6.1 Readback

Xilinx-FPGAs bieten mit dem sogenannten Readback die Möglichkeit, den gesamten internen Zustand zu jedem beliebigen Zeitpunkt über die serielle XChecker-Schnittstelle auszulesen [Höf]. Dieser Vorgang verläuft gewissermaßen spiegelbildlich zur seriellen Konfiguration, mit umgekehrter Datenflußrichtung. Er findet in zwei Phasen statt:

1. Snapshot: Sobald das spezielle Signal *RT* (für Readback Trigger) ausgelöst wird, konserviert das FPGA sein gesamtes inneres Abbild.
2. Readback: Nachdem der Snapshot angefertigt ist, kann das eigentliche Zurücklesen der kopierten Daten beginnen; sie werden seriell an *RD* (Readback Data) ausgegeben.

Mit dieser im Xilinx-Jargon *Verification* genannten Methode kann sich der Entwickler jederzeit ein Bild vom inneren Zustand der FPGAs machen. Es bestehen zwei mögliche

Erweiterungen zu diesem Grundprinzip. Das *Asynchronous Probing* realisiert durch geeignetes Auslösen des Trigger-Signals *RT* eine Hardware-Parallele zu den Breakpoints des Software-Debugging; in beiden Fällen wird der innere Zustand des Systems zu bestimmten, vordefinierten Zeitpunkten betrachtet.

Die zweite Erweiterung wird *Synchronous Probing* genannt; hier wird zusätzlich der Systemtakt durch den XChecker-Anschluß hindurchgeleitet. So kann das System, von einer Workstation aus gesteuert, nach Belieben vorübergehend angehalten werden (das Taktsignal wird dann im XChecker-Anschluß unterbrochen). Auch das Auslösen einzelner Taktschritte per Tastatur, wahlweise gefolgt von einem Readback-Prozeß, ist denkbar. Solches Vorgehen entspricht in der Software-Welt dem *Single Step*-Modus und ist sehr hilfreich, die Ursache für ein unerwartetes Systemverhalten ausfindig zu machen.

### 2.6.2 Boundary Scan

Abgesehen von den bisher behandelten Entwurfsfehlern in der FPGA-Konfiguration besteht eine ganz andere mögliche Fehlerquelle: Die Platine selbst könnte Fertigungsfehler aufweisen, etwa einen unbeabsichtigten Kurzschluß zwischen zwei Leiterbahnen. Um solche Fehler aufzudecken, gelangt häufig ein *Bed of Nails*, also „Nagelbett“ genanntes Verfahren zum Einsatz: Auf jedes Anschluß-Pad des Boards wird eine dünne Nadel gedrückt, die eine elektrische Verbindung herstellt. So können alle Verbindungen des Boards geprüft werden. Das Verfahren ist sehr aufwendig und angesichts SMD[4]- und Mehrlagentechnik immer unangemessener. Aus diesen Gründen wurde der *Boundary Scan*-Standard IEEE 1149.1 ins Leben gerufen [Par92] [AK96], der von den eingesetzten FPGAs der XC4000-Serie unterstützt wird [Xil97]. Er verfolgt einen ganz anderen Ansatz, nämlich den der Testmustergenerierung und -auswertung. Die Anschlüsse eines IC werden auf vorgegebene Pegel geschaltet, und durch Beobachten der Anschlüsse eines damit verbundenen IC lassen sich Fehler in den involvierten Leiterbahnen aufdecken.

Technisch wird diese Funktionalität wie folgt erreicht: Alle Boundary-Scan-kompatiblen ICs sind mit einem sogenannten *Test Access Port* (TAP) ausgestattet, der nichts anderes als eine weitere serielle Schnittstelle darstellt. Über den TAP können der Chip-internen Boundary-Scan-Logik Befehle erteilt werden, die sich auf die Anschlußpins des IC beziehen; der logische Pegel jedes Pins kann (ähnlich dem Readback) zurückgelesen werden, und umgekehrt kann dem TAP ein Testmuster eingegeben werden, das an die Ausgangspins des IC gelegt werden soll. Diese serielle Kommunikation kann sich per *Daisy Chain* über beliebig viele Boundary-Scan-kompatible ICs erstrecken; Daten werden also am Kopf dieser Kette eingegeben, und die Testergebnisse werden an ihrem Ende ausgelesen. Die so erhaltenen Ist-Werte werden anhand des Schaltplanes mit den Soll-Werten verglichen.

Zwar wurde der Boundary-Scan-Standard für das Aufdecken von Platinenfehlern entwickelt, doch läßt er sich in diesem Projekt auch sonst gewinnbringend ausnutzen, denn er stellt die einfachste Möglichkeit dar, einzelne oder mehrere FPGAs mit Testmustern zu versorgen. Auf diese Weise kann die Reaktion der implementierten Schaltung auf be-

[4]Surface Mounted Devices: Die Anschlußpins von Komponenten werden nicht durch Löcher in der Platine gesteckt, sondern auf der Oberfläche der Platine kontaktiert. So lassen sich wesentlich kleinere Abmessungen erzielen.

stimmte, sonst nur über Umwege herbeizuführende Situationen überprüft werden. Daher wird hier auch die Boundary-Scan-Fähigkeit der FPGAs unterstützt.

Die geschilderten Überlegungen haben ergeben, daß eine VMEbus-kompatible Auslegung des Systems als Doppeleuropa-Platine mit je vier FPGAs und Interface-Steckern sowie Konfigurations- und Arbeitsspeicher on-board sinnvoll erscheint. Die Diskussion der Entwurfsaspekte ist damit abgeschlossen; das folgende Kapitel behandelt ihre technische Umsetzung.

# Kapitel 3

# Realisierung

In diesem Kapitel wird die praktische Umsetzung der bisher erarbeiteten Überlegungen geschildert, mit dem Endergebnis des gefertigten FPGA-Boards. Die Struktur des Kapitels orientiert sich bewußt an der des vorangehenden.

## 3.1 Topologie

Die Überlegungen aus Abschnitt 2.1 führten zu der in Abbildung 3.1 dargestellten Board-internen Topologie. Die vier FPGAs sind als Array miteinander verbunden. Dabei kommen sowohl breite lokale Busse zum Einsatz, die jeweils zwei benachbarte FPGAs eng koppeln[1], als auch ein globaler Bus, auf den alle FPGAs Zugriff haben. Zusätzlich wurden die diagonal gegenüberliegenden FPGA-Paare *U1–U3* und *U2–U4* über 8-Bit-Busse miteinander verbunden (in der Abbildung aus Gründen der Übersichtlichkeit nicht dargestellt), denn es ist zu erwarten, daß in vielen Anwendungsfällen zumindest einzelne Kontroll- und Synchronisationssignale zwischen diesen Paaren ausgetauscht werden müssen. Ohne die 8-Bit-Busse müßte dies entweder über den globalen Bus geschehen, der somit unnötig belastet würde, oder die betreffenden Signale müßten durch ein drittes FPGA hindurchgeleitet werden, was einen erhöhten Verbrauch an I/O-Pins sowie eine geringere Geschwindigkeit zur Folge hätte.

Jedes der vier FPGAs hat exklusiven Zugriff auf einen eigenen 96poligen Stecker nach außen. Zwei dieser Stecker können als Schnittstelle zum VMEbus genutzt werden, die anderen beiden sind auf der gegenüberliegenden Seite des Boards als Komplemente dazu ausgeführt, so daß mehrere FPGA-Boards (auf Wunsch auch seitlich versetzt) hintereinandergeschaltet werden können. Somit sind globale Topologien, insbesondere Gitternetze wie in Abbildung 3.2 realisierbar. Ähnlich dem WEAVER-System in Abbildung 1.5 können mittels einfacher Bus-Platinen dreidimensionale Strukturen realisiert werden.

In jedem Fall besteht die Option, nachgeschaltete Boards über die Verbindungsstecker mit dem Taktsignal sowie mit der Betriebsspannung zu versorgen, es sind also keine weiteren externen Versorgungsleitungen dafür erforderlich.

[1] Diese enge Kopplung fällt zwischen *U3* und *U4* wegen des involvierten *Dual Port*-Speichers indirekter aus, s. u.

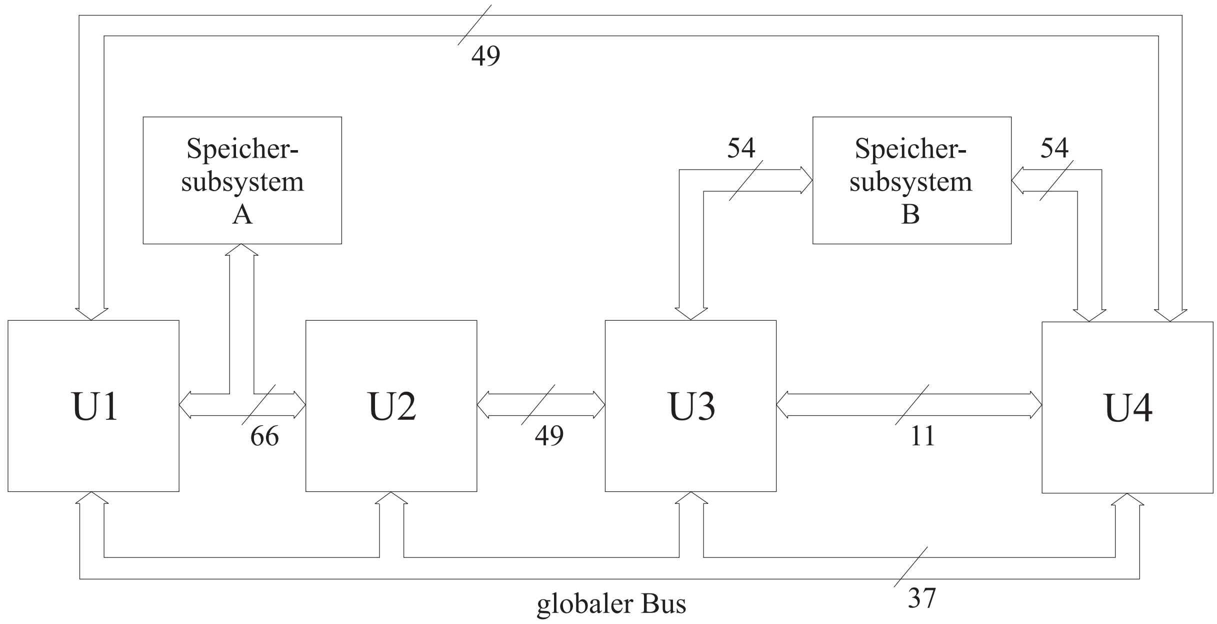


Abbildung 3.1: Vereinfachte Darstellung der Board-Topologie

In der Abbildung ist weiterhin zu erkennen, daß die FPGAs *U1* und *U2* Zugriff auf gemeinsamen Speicher haben; das gleiche gilt für *U3* und *U4*. Hierauf wird in Abschnitt 3.3 näher eingegangen.

## 3.2 FPGAs

Bei der Umsetzung des hier vorgestellten Prototyping-Systems stand früh fest, daß FPGAs der Xilinx XC4000-Serie zum Einsatz kommen sollten. Ihre SRAM-basierten Konfigurationsressourcen sind besonders geeignet für häufiges Neuprogrammieren der Chips, und insbesondere das geplante Leistungsmerkmal der dynamischen Rekonfiguration ist mit anderen Technologien nicht umsetzbar. Mit XC4000-FPGAs wurden am Lehrstuhl viele Erfahrungen gesammelt, und eine komplette Entwicklungsumgebung ist vorhanden.

Zum Zeitpunkt der Beschaffung der FPGAs boten die leistungsstärksten Typen der XC4000-Serie Kapazitäten bis ungefähr 85.000 Gatteräquivalenten, so daß eines oder zwei dieser Modelle ausgereicht hätten, die angestrebte Kapazität von ca. 100.000 Gattern je Board zu erbringen. Allerdings fließt auch die Anzahl der I/O-Pins je Chip stark in die Überlegungen der Auswahl ein: so bieten vier FPGAs mit je 299 Pins unter Umständen eine bessere Konnektivität für externe Komponenten als ein einzelnes mit 559 Anschlüssen. (Die genannten Zahlen beziehen sich auf lieferbare PGA[2]-Versionen der Chips. Der Einsatz des PGA-Gehäuses steht außer Frage, denn die anderen Gehäuseformen lassen sich sich nicht sockeln. Sockel sind erforderlich, um die FPGAs entnehmen und beispielsweise durch leistungsfähigere Nachfolgemodelle oder auch durch einen Mikroprozessor austauschen zu können. Das derzeit größte erhältliche PGA-Gehäuse bietet 559 Anschlüsse.)

[2]Pin Grid Array: eine Matrix konventioneller Anschlußpins, die durch in die Platine gebohrte Löcher montiert werden, im Gegensatz zur Oberflächenmontage

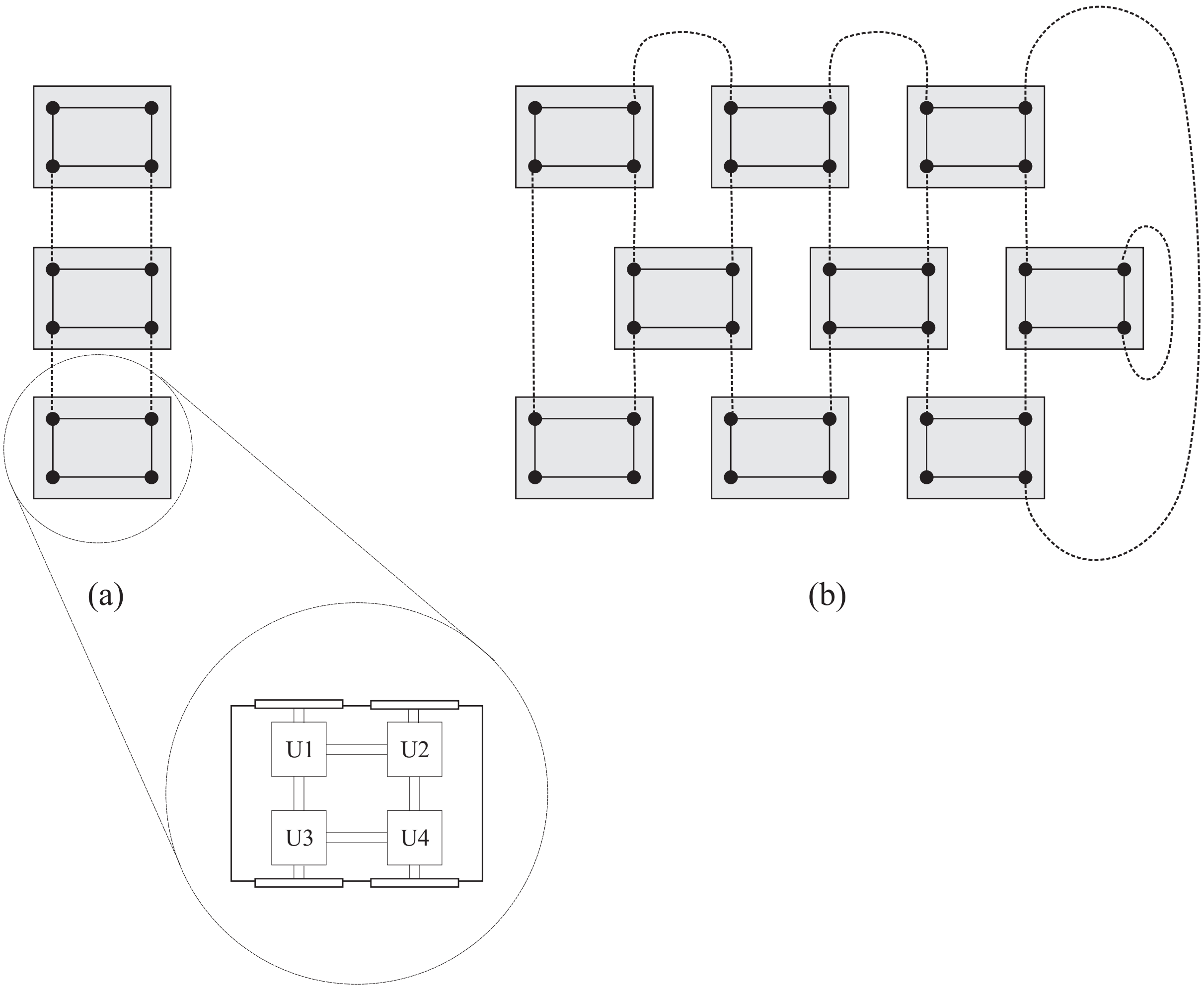


Abbildung 3.2: FPGA-Boards im Verbund: (a) lineare Hintereinanderschaltung, (b) Gitternetz

Außerdem stellt die Verdrahtung von ICs mit 559 Pins hohe Anforderungen an die Platine, und schließlich beeinflußt die Anzahl der Anschlußpins den Preis der FPGAs stärker noch als die Anzahl ihrer CLBs. Die geeignetste Wahl schien nach Abwägung der geschilderten Fakten und unter Berücksichtigung der Überlegungen aus Abschnitt 2.1.2 der Einsatz von vier FPGAs mit je 299 Pins zu sein, und innerhalb dieser Vorgabe die Variante mit der größten logischen Kapazität; das ist derzeit das Modell XC4028 mit etwa 28.000 Gatteräquivalenten.

## 3.3 Speicher

### 3.3.1 Speichersubsystem A

In Abbildung 3.1 sind zwei Speichersubsysteme zu erkennen. Subsystem A setzt sich zusammen aus Konfigurations- und *Single Port*-Speicher; an dieser Stelle wird näher beleuchtet, wie diese implementiert sind. (Subsystem B besteht allein aus Dual-Port-Speicher, der weiter unten behandelt wird.) Die einzelnen Komponenten des Speichersubsystems A

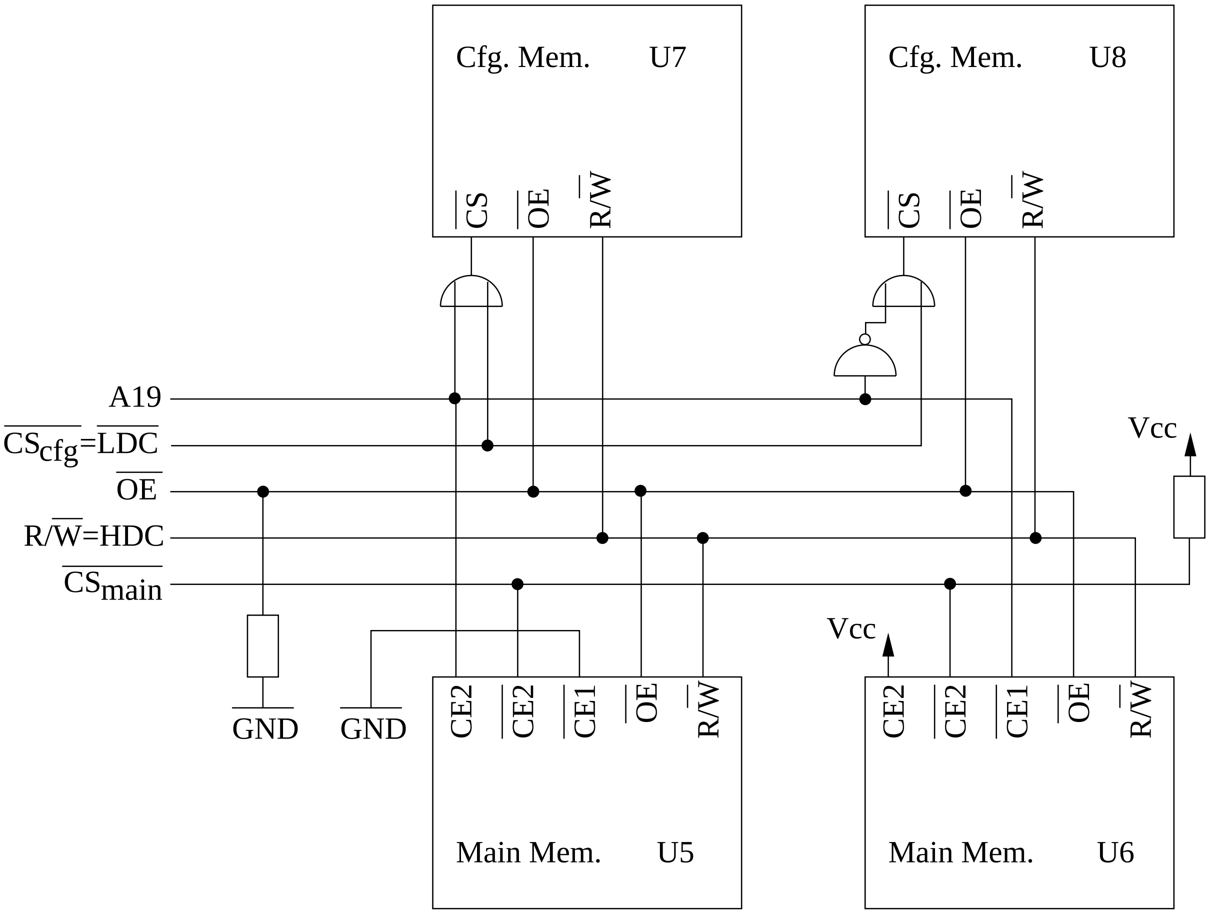


Abbildung 3.3: Die Komponenten des Speichersubsystems A

sind in Abbdilung 3.3 dargestellt. Konfigurations- und Arbeitsspeicher bestehen jeweils aus zwei Chips, die mit dem Adreßsignal $A19$ wechselweise ausgewählt werden, so daß der Adreßraum verdoppelt wird. Er umfaßt in beiden Fällen $2^{20}$ Wörter.

Während der Konfigurationsphase sind die meisten der FPGA-Pins hochohmig, mit Ausnahme einiger spezieller Konfigurationssignale. Es muß gewährleistet sein, daß die Konfigurationsspeicher-Eingänge $\overline{CS}$[3] und $\overline{OE}$[4] auf Low-Pegel liegen, während Eingang $R/\overline{W}$[5] einen High-Pegel erfordert. In der Abbildung ist zu erkennen, wie dieser Zustand mittels geeigneter Verwendung der von *U1* bereitgestellten Signale $\overline{LDC}$ und $HDC$, die während der Konfiguration einen Low- bzw. High-Pegel liefern und danach frei programmierbar sind, erreicht wird.

Der Arbeitsspeicher ist während der Konfigurationsphase deaktiviert, da das Signal $CS_{main}$ über einen Pull-Up-Widerstand auf hohes Potential gezogen wird. Nach Abschluß der Konfigurationsphase sind alle dargestellten Signale benutzerprogrammierbar, so daß wahlweise auf den Konfigurations- oder den Arbeitsspeicher zugegriffen werden kann. Es

[3] Chip Select
[4] Output Enable
[5] Read/Write

ist auch möglich, beide Speicher zu deaktivieren und den Bus des Speichersubsystems A stattdessen für die Kommunikation zwischen *U1* und *U2* zu verwenden (vgl. Abbildung 3.1).

Im folgenden werden die Komponenten der Speichersubsysteme A und B einzeln eingehend behandelt.

### 3.3.2 Konfigurationsspeicher (*U7*, *U8*)

Nachdem in Abschnitt 2.3 der Nutzen eines nichtflüchtigen Speichers für Konfigurationsdaten dargestellt wurde, soll hier auf dessen Realisierung eingegangen werden. Seine Einbindung in die Kommunikationsstrukturen des Boards wird weiter unten zusammen mit dem *Single Port*-RAM erläutert, und der Konfigurationsvorgang an sich wird schließlich in Abschnitt 3.7 beschrieben. An dieser Stelle genügt es zu wissen, daß das Master-FPGA *U1* im hier relevanten *Master Parallel*-Modus selbständig Adressen generiert, mit denen es ein Datenarray von 8 Bit Breite asynchron ausliest und optional weitere, im Slave-Modus betriebene FPGAs mitkonfiguriert. (Für Details vgl. [Alf97] [Xil99] [Xil96].) Je FPGA werden knapp 82 kByte an Daten benötigt, etwa 327 kByte für alle vier. Entsprechend mehr Speicher könnte sinnvoll sein, wenn die Konfigurationskette um weitere Boards ergänzt werden soll, die dann über keinen eigenen Konfigurationsspeicher verfügen müssen. Auch für das Halten alternativer Konfigurationen, die das Master-FPGA *U1* zur Laufzeit in nachgeschaltete FPGAs schreiben kann (vgl. Abschnitt 3.7.2), ist ein Mehr an Speicher nützlich.

#### Denkbare Varianten der Implementierung

Prinzipiell stehen verschiedene Optionen für die Realisierung des nichtflüchtigen Konfigurationsspeichers zur Wahl:

- EEPROM[6]: Das Master-FPGA stellt beim Auslesen des Konfigurationsspeichers unkritische Timing-Anforderungen, so daß die relativ lange Zugriffszeit von EEPROMs kein Hindernis darstellt. Allerdings ist EEPROM nur für seltene Schreibzugriffe ausgelegt und geeignet; dem bloßen Einsatz als Konfigurationsspeicher steht das nicht im Wege, aber eine optionale, zusätzliche Verwendung zur dynamischen Datenhaltung scheidet gänzlich aus.

- Flash-EPROM: Hauptvorteil dieser weiterentwickelten EEPROM-Technologie ist der größere realisierbare Speicher; Ausführungen mit 1 MByte und mehr sind durchaus erhältlich. Erkauft wird dieser Vorteil allerdings mit einer recht umständlichen und noch dazu hersteller-abhängigen, sektor-orientierten Programmierung Das Flash-EPROM kann nur mit einigem Aufwand *in situ*, d. h. in montiertem Zustand auf dem Board programmiert werden die externe Programmierung erfordert ein geeignetes Programmiergerät und das Entnehmen des Bauelementes. In jedem Fall

[6] Electrically Erasable and Programmable ROM; vorwiegend für den Lesezugriff konzipierter nichtflüchtiger Speicher, der aus Anwendersicht ähnliche Schreibzugriffe gestattet wie asynchrones SRAM, allerdings mit um Größenordnungen geringerer Geschwindigkeit. Die Anzahl der Schreibzugriffe je Speicherzelle ist stark beschränkt; etwa $10^4$ Zyklen sind eine übliche Lebensdauer.

kann es notwendig werden, herstellerabhängige Programmieralgorithmen zu implementieren.

- NV-RAM: *Non Volatile RAM*, also „nichtflüchtiges RAM“, wurde mit dem Ziel entwickelt, die Datenerhaltung auch bei abgeschalteter Versorgungsspannung mit dem unkomplizierten, schnellen Lese- und Schreibzugriff gewöhnlichen SRAMs zu kombinieren. Es gibt verschiedene Varianten der Implementierung. Für kleinere Kapazitäten werden ein normales SRAM- sowie parallel dazu ein EEPROM-Die[7] mit einer im Moment des Spannungsausfalls aktivierten Backup-Logik in einem IC-Gehäuse untergebracht. In anderen Fällen wird zusätzlich zu einem SRAM-Die schlicht eine Batterie in das Gehäuse integriert. Unabhängig von der internen Realisierung entsteht für den Anwender der Eindruck eines nichtflüchtigen, davon abgesehen aber völlig normalen RAM.

  Leider stellt NV-RAM in der hier benötigten Größenordnung ein relativ wenig nachgefragtes Spezialprodukt dar, das von wenigen Distributoren angeboten wird und sich zudem häufig noch in der Anfangsphase der Produktion befindet [Dal98]. In geringer Stückzahl oder auch als Muster war es nicht erhältlich.

- *Smart Sockets:* Einige Hersteller offerieren spezielle Sockel, die Standard-SRAM-ICs mittels integrierter Batterie und automatischer Spannungsquellen-Umschaltung scheinbar in nichtflüchtigen Speicher verwandeln. Diese Lösung ist dem echten NV-RAM sehr ähnlich und bietet die gleichen Vorteile, allerdings auch den gleichen Nachteil der schwierigen Beschaffung, insbesondere als 3,3-V-Variante.

- gepuffertes SRAM: Der NV-RAM-Effekt mit allen seinen Vorteilen läßt sich auch unter Zuhilfenahme separater Komponenten erzielen; dieser Ansatz stellt im übrigen die preiswerteste Variante dar. Benötigt wird eine Logik, die das Wechseln zwischen Haupt- und Batteriespannungsversorgung vornimmt und bei letzterer das *Chip Enable*-Signal deaktiviert, um den Stromverbrauch zu reduzieren und den Inhalt des SRAM vor versehentlichen Schreibzugriffen zu schützen. Für diese Aufgabe sind spezielle integrierte Schaltungen erhältlich, die als *Non Volatile SRAM Controller* oder *Microprocessor Supervisory Circuit* bezeichnet werden. Davon abgesehen ist noch die eigentliche Energiequelle vorzusehen; prädestiniert sind aufgrund ihrer hohen Lebensdauer Lithium-Batterien oder alternativ spezielle, sogenannte *Gold Cap*-Kondensatoren mit besonders großer Kapazität.

### Batterie-Pufferung

Die Entscheidung gegen EPROM und somit für eine doppelte Nutzbarkeit als Konfigurations- wie als Arbeitsspeicher fällt angesichts dieser Sachlage leicht. Das Board läßt sich mit NV-RAM betreiben; angesichts der erwähnten Beschaffungsschwierigkeiten bei diesem Speichertyp wurde das System aber zusätzlich für den Einsatz des *Non Volatile SRAM Controllers* MAX795 der Firma Maxim vorbereitet [Max96]; für seine Beschaltung vgl. *IC1* in Abbildung B.3: Bei Vorhandensein der Betriebsspannung *VCC* wird sie auf den Ausgang *OUT* durchgeschaltet, anderenfalls wird dieser mit der Batteriespannung

[7] Die: Silizium-Plättchen von der Größe weniger Quadratmillimeter; Kern jedes IC

*BATT* gespeist (diese drei Spannungsversorgungspins sind mit den Kondensatoren *C92* bis *C94* gepuffert). Solange die Betriebsspannung *VCC* anliegt, wird das *Chip Enable*-Signal transparent durchgeschleift; andernfalls wird es auf hohen Pegel gezogen, um das RAM vor unkontrollierten Schreibzugriffen zu schützen, insbesondere während des Ein- und Ausschaltens der Hauptstromversorgung.

### Dimensionierung

Die größten derzeit erhältlichen SRAM-ICs bieten eine Kapazität von 512 kByte; das Board ist für bis zu zwei solcher Bauelemente ausgelegt, deren Gesamtkapazität von 1 MByte für 12 Konfigurationen ausreicht. Zusätzlich zu den für ein Board erforderlichen vier Konfigurationen können also mehrere weitere vorgehalten werden, die etwa für die dynamische Rekonfigurierung einiger FPGAs oder für das Versorgen zweier weiterer, nachgeschalteter Slave-Boards verwendet werden können. (Selbstverständlich können auch größere Verbunde realisiert werden, nur sind dann weitere Quellen für Konfigurationsdaten erforderlich — seien es weitere mit Konfigurationsspeicher bestückte Boards, oder sei es eine Workstation mit XChecker-Ausgang; weitere Informationen finden sich in Abschnitt 3.7.)

### Externer Zugriff auf den Konfigurationsspeicher

Nun muß noch das Problem angegangen werden, wie die Konfigurationsdaten in den Speicher geschrieben werden können. Ein gangbarer Weg ist es, *U1* oder *U2* über das jeweilige XChecker-Interface so zu konfigurieren, daß sämtliche Anschlüsse des Konfigurationsspeichers transparent durch das betreffende FPGA an den zugehörigen Außenstecker *ST1* bzw. *ST2* durchgeschaltet werden. Daraufhin besteht volle Zugriffsmöglichkeit auf den Speicher von außen, er kann mit den Konfigurationsdaten programmiert werden, und fortan steht er als Konfigurationsdatenquelle für das Master-FPGA *U1* zur Verfügung.

Einfacher wäre es natürlich, wenn auf den Speicher unmittelbar über ein eigenes Interface nach außen zugegriffen werden könnte. Zu diesem Zweck wurde die ebenfalls in Abbildung B.3 dargestellte 36polige Stiftleiste *JP21* vorgesehen, die Zugriff auf sämtliche Anschlüsse des Konfigurationsspeichers gewährt. Da dieser zusammen mit dem *Single Port*-Speicher sowie den FPGAs *U1* und *U2* an einem gemeinsamen Bus arbeitet, muß eine Möglichkeit vorgesehen werden, die übrigen Busteilnehmer während des externen Zugriffs auf den Konfigurationsspeicher zu deaktivieren. Zu diesem Zweck wird das spezielle, globale Signal *GTS* eingesetzt [Xil99], das die zeitweilige Deaktivierung aller I/O-Pins der vier FPGAs erlaubt. In dieser Situation deaktiviert der Pull-Up-Widerstand *R9* die *Single Port*-Speicherbausteine, sobald der nächste Taktschritt ausgeführt wird (vgl. Abbildung B.4). Das bedeutet, daß *JP21* bei aktiviertem *GTS*-Signal exklusiven und ungehinderten Zugriff auf den Konfigurationsspeicher bietet. (Alternativ können die FPGAs auch einfach unkonfiguriert belassen werden, während von außen auf *JP21* zugegriffen wird. Dazu genügt es, *U1* als Slave zu betreiben und keine Konfigurationsdaten anzulegen.)

### 3.3.3 Arbeitsspeicher

#### *Single Port*-Speicher (*U5*, *U6*)

An dieser Stelle wird der *Single Port*-Speicher behandelt; wie gesehen, bildet er zusammen mit dem Konfigurationsspeicher das Subsystem A. Durch seine Einbindung in den Bus zwischen *U1* und *U2* bietet sich den genannten FPGAs die Möglichkeit, wahlweise mit dem jeweils anderen FPGA oder mit einem der Speicher des Subsystems A zu kommunizieren. Bei getrennter Zuteilung von jeweils lokalem Speicher an beide FPGAs stünden keine Leitungen mehr für die Kommunikation zwischen ihnen zur Verfügung. Es werden also I/O-Ressourcen eingespart mit dem Preis, daß sich die FPGAs den Zugriff auf das Speichersubsystem A teilen müssen.

Der hier behandelte *Single Port*-Speicher unterscheidet sich in zwei Aspekten vom Konfigurationsspeicher, den unterschiedlichen Anforderungen entsprechend:

- Synchrone Arbeitsweise: Im Gegensatz zum Konfigurationsspeicher kann hier synchrones RAM eingesetzt werden, das den schnelleren und einfacheren Zugriff bietet. Es müssen keine Read- und Write-Strobe-Pulse generiert werden, die die Gültigkeit der anliegenden Signale anzeigen, da das Taktsignal diese Funktion übernimmt.
- Wortbreite: Natürlich besteht ein linearer Zusammenhang zwischen dem Datendurchsatz und der Anzahl der pro Takt übertragenen Datenbits, was das Bestreben nahelegt, die Wortbreite möglichst groß zu wählen. Eine absolute Obergrenze ergibt sich dabei durch die begrenzte Anzahl zur Verfügung stehender FPGA-Pins, doch auch davon abgesehen sprechen praktische Einwände gegen allzu große Werte: Je breiter der Datenbus, desto höher ist die Wahrscheinlichkeit, daß die jeweilige Anwendung nicht alle Datenbits ausnutzt; folglich würden sowohl FPGA-Pins als auch Speicherkapazität vergeudet. Auch hier handelt es sich also um die Suche nach einer sinnvollen Granularität, die immer mit dem Abwägen konkurrierender Ziele verbunden ist.

  Aufgrund der grob umrissenen Board-Topologie war früh abzusehen, daß jeweils etwa ein Viertel der ca. 250 I/O-Pins der FPGAs *U1* und *U2* für den *Single Port*-Arbeitsspeicher zur Verfügung stehen würde; subtrahiert man davon als Richtwert 5 Kontroll- und 20 Adreßleitungen, was einem Adreßraum von $2^{20} \approx 1$ Mio. Wörtern entspricht, so verbleiben etwa 40 verfügbare Signale für die Datenbits als absolute Obergrenze. Eine größere Wortbreite wäre ohnehin nicht sinnvoll erschienen: Für die meisten Zwecke dürfte sie ausreichend sein, und für darüber hinausgehende Anforderungen besteht selbstverständlich die Möglichkeit, jeden Transfer in mehrere Taktzyklen zu zerlegen.

Speicherbausteine, die den genannten Anforderungen genügen, werden derzeit mit einer ausreichenden Kapazität angeboten. Mit erzielbaren Taktfrequenzen von 100 MHz und mehr werden sie häufig als Cache[8] eingesetzt. Ihre hohe Geschwindigkeit erlaubt hier den Zugriff ohne *Wait States*. Um die Performance zu maximieren, haben die Halbleiterhersteller in jüngerer Zeit immer neue Varianten ins Leben gerufen, darunter *Pipelined Burst*

[8] besonders schneller Zwischenspeicher

und *Late Write SRAM*. Als neueste Entwicklung führten IDT, Micron und Motorola das *ZBT*[9] *SRAM* ein, den ersten synchronen Speicher, der keine ungenutzten Taktzyklen beim Wechsel zwischen Schreib- und Lesezugriffen erfordert. In Anwendungen mit langen Folgen gleichartiger Zugriffe (sog. *Bursts*), wie sie zum Beispiel in Cache-Systemen vorliegen, machen die ungenutzten Taktzyklen nur einen vernachlässigbaren Anteil aus; für eingebettete Systeme aber, und somit auch für das hier entwickelte Prototyping-System, das eingebettete Systeme simulieren soll, gilt das im allgemeinen nicht: Die Zugriffe können beliebig verzahnt auftreten. Im ungünstigsten Fall des alternierenden Lesens und Schreibens bleiben beim *Pipelined Burst SRAM* 50 % der Taktzyklen ungenutzt. Eine Übersicht und Besprechung der Speichertechnologien findet sich in [Bur98] und [Mic98].

Häufig wird die Leistungsfähigkeit elektronischer Komponenten durch Fließbandverarbeitung, das sog. *Pipelining* erhöht: Auszuführende Aktionen werden in mehrere Phasen zerlegt; noch während sich eine Aktion in einer fortgeschrittenen Phase der Bearbeitung befindet, kann die erste Phase der nächsten Aktion ausgeführt werden. ZBT-RAM ist mit und ohne *Pipelining* erhältlich. Da sie pin-kompatibel sind, werden beide Varianten vom Board unterstützt; am geeignetsten erscheint jedoch die *Flow Through* genannte Variante ohne *Pipelining*, denn sie liefert bzw. erhält die Daten mit einer Verzögerung von nur einem Takt nach der Adresse, während das *Pipelined ZBT RAM* zwei Takte Verzögerung aufweist, was die Handhabung seitens der FPGAs unnötig erschwert. Ohnehin erlaubt selbst die *Flow Through*-Ausführung Taktfrequenzen von 100 MHz und dürfte damit oberhalb der Grenzen liegen, die dem Prototyping-System durch die FPGAs gesetzt sein werden — ganz abgesehen von der Platine an sich, denn derartige Frequenzen stellen extreme Anforderungen an ihr Layout, etwa was die maximale Leitungslänge angeht.

Die genannten Fakten führten zu der Entscheidung, das Board für den Einsatz von bis zu zwei synchronen ZBT-SRAMs auszulegen und es mit zwei *Flow Through*-Komponenten in der momentan größten erhältlichen Ausführung von 128k x 36 Bit zu bestücken [IDT98d]. Die für die Zukunft angekündigten Speicherbausteine mit Kapazitäten von bis zu 512k x 36 Bit [Mot99] werden dank entsprechender Beschaltung der beiden reservierten Adreß-Pins bereits unterstützt; das führt zu einem maximalen Adreßraum von $2^{20} \approx 1$ Mio. Wörtern, identisch mit dem Konfigurationsspeicher. Den FPGAs *U1* und *U2* stehen somit maximal 36 MBit an Arbeits- sowie 8 MBit an Konfigurationsspeicher zur gemeinsamen Verfügung.

### ***Dual Port*-Speicher (*U9*, *U10*)**

Der *Dual Port*-Speicher bildet das Speichersubsystem B. Auch hier wurde ein Konzept umgesetzt, um zwei FPGAs (nämlich *U3* und *U4*) mit Speicher auszurüsten, die dafür aufgewendeten Leitungen aber gleichzeitig für die Kommunikation beider FPGAs untereinander einsetzen zu können. Diese Kommunikation läuft über den Speicher ab; statt Daten von FPGA zu FPGA zu senden, schreibt sie der Sender in den Speicher, und der Empfänger liest sie später aus. Weil der Speicher zwei Ports hat, behindern sich die FPGAs dabei nicht; sie können gleichzeitig auf ihn zugreifen. Ein einzelner Datentransfer benötigt hier mehrere Taktzyklen (zum Schreiben und anschließenden Lesen), was sich

[9] Zero Bus Turnaround, etwa „Nullzeit-Busrichtungswechsel“; von anderen Herstellern werden kompatible Produkte mit Namen wie „No Bus Latency“ angeboten

nachteilig auswirkt, wenn die Kommunikation zwischen den FPGAs sehr häufig die Richtung wechselt. Für *Burst*-Übertragungen aber, also Folgen von Transfers in der gleichen Richtung, besteht praktisch kein Performance-Unterschied zur direkten Kommunikation.

Der Hauptvorteil der Kommunikation über *Dual Port*-Speicher ist die zeitliche Entkopplung. Der Sender muß nicht warten, bis der Empfänger bereit ist zur Annahme der Daten. Derartige Systeme lassen sich gewinnbringend einsetzen, wenn zwei Prozessoren verschiedene, aber voneinander abhängige Arbeiten verrichten [IDT98a]. Neben diesem Kommunikationsmechanismus kann dasselbe *Dual Port*-RAM natürlich von jedem der beiden FPGAs ganz konventionell zur Datenhaltung genutzt werden.

Das im Zusammenhang mit dem *Single Port*-Arbeitsspeicher Gesagte über Wortbreite und synchrone Arbeitsweise gilt auch hier. Weil *Dual Port*-Speicher wesentlich aufwendiger in der Herstellung sind, sind sie derzeit nur in Wortbreiten bis 16 Bit erhältlich, daher wurden zwei Chips „nebeneinander“ geschaltet. Auch ihre Kapazität ist relativ gering; im Rahmen dieser Arbeit wurden mit 32k x 16 Bit die größten derzeit erhältlichen Chips eingesetzt. Schließlich sind *Dual Port*-Speicher (noch?) nicht in ZBT-Ausführung erhältlich. Beim Wechsel zwischen Lese- und Schreibzugriffen (auf denselben Port) gehen daher im Falle von Speichern mit Pipeline-Architektur zwei Taktzyklen verloren, bei Flow-Through-Architekturen einer. Dafür sind Pipeline-Versionen derzeit bis 66 MHz erhältlich [IDT98c], Flow-Through-Varianten nur bis 33 MHz [IDT]. Es hängt von der vorgesehenen Einsatzweise der Speicher ab, welche Variante geeigneter ist.

## 3.4 Interfaces

### 3.4.1 VMEbus-Betrieb

Um das Prototyping-Board kompatibel zum VMEbus-Standard [VME95][Pet97] zu machen, stehen grundsätzlich zwei Varianten zur Verfügung: einerseits die Verwendung eines dedizierten VME-Chipsatzes [Tun98][Cyp97a][Cyp97b], andererseits die Implementierung des geforderten Protokolls unter dem Einsatz von FPGA-Logik. Die Verwendung eines dedizierten Chipsatzes wurde in Betracht gezogen, schied aber aufgrund der Komplexität solcher Lösungen aus. Dies sei am Beispiel des Cypress VIC64 verdeutlicht: Dieser Chip hat mindestens 144 Anschlußpins und erfordert drei bis vier weitere unterstützende Chips des Typs CY7C964, die wiederum jeweils mindestens 64 zusätzliche Anschlußpins beisteuern. Dazu kommen noch weitere Bauteile wie Taktgeber u. a., so daß ein solches Interface eine beträchtliche Platinenfläche in Anspruch nimmt. Diese Fläche wäre ausschließlich im Falle des VMEbus-Betriebes sinnvoll genutzt, und selbst bei Verwendung eines solchen speziellen Chipsatzes wird weiterhin viel FPGA-Logik für seine auf diversen Registern, Mailboxen und Interrupts beruhende Ansteuerung benötigt.

Der größte Nachteil einer Lösung mit VME-Chipsatz ist aber der, daß die betreffenden Interfaces zwangsläufig permanent mit dem VME-Protokoll betrieben werden müßten. Somit stünden sie nicht mehr für die Verbindung mehrerer Boards untereinander zur Verfügung.

Daher wurde die Entscheidung getroffen, die FPGAs unmittelbar mit den VMEbus-Steckern zu verbinden, das Busprotokoll muß also ausschließlich mittels der FPGA-Logik implementiert werden, doch dies erscheint unter den gegebenen Bedingungen mehr als vertretbar.

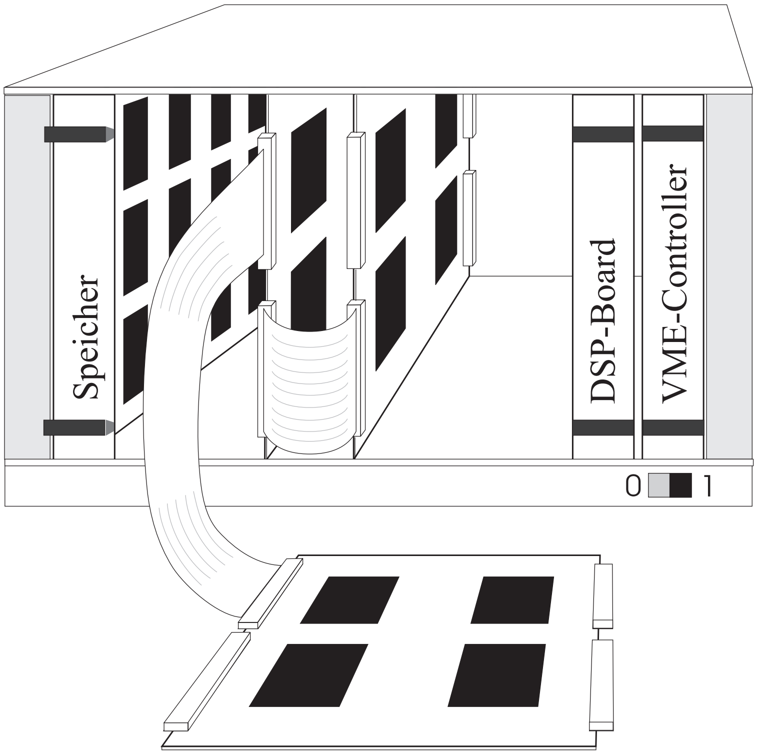


Abbildung 3.4: VME-Rack mit einer Beispielkonfiguration eingesetzter Hardware

Im VMEbus-Betrieb stehen selbstverständlich die beiden Interfaces *U3* und *U4* an der Frontseite der Platine weiterhin zur freien Verfügung. So können beispielsweise mehrere im VME-Rack betriebene FPGA-Boards untereinander zu einem Ring verschaltet werden (ähnlich dem Grundkonzept des in Abschnitt 1.4.2 vorgestellten ArMen-Systems), oder sie können mit externer Hardware kommunizieren. Diese Optionen sind beispielhaft in Abbildung 3.4 dargestellt.

### 3.4.2 Verbund-Betrieb

Im vorigen Abschnitt wurde beschrieben, wie das FPGA-Board über das VME-System mit anderen Hardware-Komponenten gekoppelt werden kann; derartigen Gesamtsystemen liegt immer die Bus-Topologie zugrunde. Im Gegensatz dazu wird nun die unmittelbare Verbindung mehrerer FPGA-Boards miteinander behandelt, die auch andere Topologien ermöglicht. Grundvoraussetzung für dieses Leistungsmerkmal sind die beiden DIN 41612-R-kompatiblen Buchsen auf der Frontseite der Platine, die das genaue Gegenstück zu den DIN 41612-C-kompatiblen VME-Steckern darstellen. Mittels dieser Interfaces ist es ohne weiteres möglich, mehrere FPGA-Boards linear hintereinanderzuschalten, wie in Abbildung 3.2 (a) gezeigt. Hier sind immer die FPGAs *U1* mit *U3* verbunden, und *U2* mit *U4*. Es genügt also sicherzustellen, daß die Pinbelegung der Buchse *ST3* mit der des Steckers *ST1*, die der Buchse *ST4* mit der des Steckers *ST2* harmoniert.

Für komplexere Strukturen wie in Teil (b) der Abbildung allerdings ist es unerläßlich, daß auch die Paare *ST2*–*ST3* sowie *ST1*–*ST4* miteinander gekoppelt werden können. Dem steht die VMEbus-Spezifikation entgegen, denn dort sind für *ST1* und *ST2* inkompatible Belegungen der Spannungsversorgungspins vorgesehen. Als Lösung dieses Konfliktes lassen sich die VMEbus-Stecker des FPGA-Boards mittels einiger Jumper an die jeweils angeschlossene Komponente (VMEbus oder ein anderes FPGA-Board) anpassen. Dabei wurde beachtet, daß in jeder Situation genügend Versorgungs-Anschlüsse zur Verfügung stehen, und sofern es sich mit dem VMEbus-Standard vereinbaren ließ, wurden die Versorgungssignale gleichmäßig über die Breite der Interfaces verteilt, um die elektrische Entkopplung der Datensignale untereinander zu optimieren.

Im Verbund-Betrieb können Spannungs- und Taktversorgung auf Wunsch transparent und VMEbus-kompatibel von den Steckern *ST1* und *ST2* an die Buchsen *ST3* und *ST4* weitergeleitet werden. Auch soll auf die Möglichkeit hingewiesen werden, Paare von Interfaces eines einzelnen Boards miteinander zu verbinden, so daß die *interne* Konnektivität verbessert wird. Durch kreuzweises Verbinden der Interfaces beispielsweise lassen sich die vier FPGAs eines Boards vollständig vermaschen.

## 3.5 Spannungsversorgung

### 3.5.1 5 V oder 3,3 V?

Die überwiegende Mehrheit der elektronischen Schaltungen arbeitet mit einer Versorgungsspannung von 5 V, und noch dürften die meisten erhältlichen Bauelemente für diese Spannung ausgelegt sein, doch der Trend geht zu kontinuierlich niedrigeren Pegeln, da sie Vorteile hinsichtlich Schaltungsgeschwindigkeit (ein geringerer Spannungsunterschied bedeutet eine kürzere Pegelübergangszeit) und Leistungsaufnahme bieten. (Die niedrigere Versorgungsspannung läßt weniger Strom fließen; beides bedeutet eine verringerte Leistungsaufnahme. Damit wird weniger Verlustwärme produziert und eine höhere Integrationsdichte der mikroelektronischen Schaltkreise ermöglicht.) Für fundierte Einschätzungen der zukünftigen Entwicklung der Halbleitertechnologie sei auf den periodisch aktualisierten Bericht der Semiconductor Industry Association verwiesen [SIA97].

Auch in der vorliegenden Arbeit wurde auf die *Low Voltage*-Technologie zurückgegriffen, zumal die 3,3-V-Varianten der ins Auge gefaßten FPGAs der Firma Xilinx (vgl. Abschnitt 3.2) zu den genannten Vorteilen noch einen um 40 % niedrigeren Preis auf sich vereinen.

### 3.5.2 Konflikt mit VMEbus-Pegeln

In Abschnitt 2.4 wurde der Beschluß erwähnt, das Board VME64-kompatibel zu gestalten. Weil dieser aber ausschließlich mit Pegeln von 5 V arbeitet, während die Eingangspegel von ICs nicht „wesentlich" (etwa um 0,5 V) oberhalb ihrer Versorgungsspannung liegen dürfen, ist ein Konflikt zwischen den beiden konkurrierenden Entwurfsentscheidungen zugunsten der VMEbus-Kompatibilität und der *Low Voltage*-Technologie zu erwarten. Auf letztere wäre verzichtet worden, wenn sich dedizierte Pegelübersetzer-Bausteine zwischen den

VMEbus-Anschlüssen und den FPGAs als unverzichtbar erwiesen hätten. Dieser Mehraufwand hätte nicht mehr im gewünschten Verhältnis zum Nutzen gestanden, zumal fast alle Signale auf dem Board bidirektional arbeiten; somit wäre es erforderlich geworden, den Pegelübersetzer-Treibern die Richtung jedes Signalflusses anzuzeigen. Diese Richtungssignale wären für die logische Funktionalität des Boards verloren gewesen, ihre Generierung hätte einen Mehraufwand an Logik bedeutet, und schließlich wäre Flexibilität, jede einzelne Leitung jederzeit in einer beliebigen Richtung betreiben zu können, auf vordefinierte Gruppen von Leitungen eingeschränkt worden.

Die 5-V-toleranten FPGA-Eingänge sowie besondere Vorkehrungen bei der Taktverteilung aber ermöglichen den Verzicht auf jegliche Pegelübersetzer:

- Übersetzung der VMEbus-Pegel durch FPGAs: Die gewählten FPGAs der Xilinx XL-Serie arbeiten zwar mit einer Spannung von 3,3 V, aber ihre Eingänge sind aufgrund spezieller technischer Maßnahmen *5-V-tolerant* (vgl. [AC97] für weitere Informationen); diese FPGAs bilden also eine Ausnahme von der Regel, daß Eingangssignale eines IC nicht deutlich oberhalb ihrer Versorgungsspannung liegen dürfen. Wie aus Abschnitt 3.1 ersichtlich, ist abgesehen vom Systemtakt kein VMEbus-Signal mit einem anderen Bauteil verbunden als mit FPGAs. Es bleibt also nur noch Sorge zu tragen, daß das Taktsignal an die 3,3-V-Pegel des Boards angepaßt wird.

- Geeignete Wahl der Takt-Multiplexer und -Treiber: Hierauf wird im Zusammenhang mit der Taktverteilung in Abschnitt 3.6 eingegangen.

Darüber hinaus sind nur wenige weitere Maßnahmen erforderlich, um das Prototyping-System in der *Low Voltage*-Technologie zu realisieren:

- Spannungsregler: Die 5-V-Versorgungsspannung, die das VME-Rack an alle eingesetzten Karten liefert, muß mittels eines Spannungsreglers auf 3,3 V begrenzt werden. Er bildet das Bindeglied zwischen dem $VCC$- und dem $5V/3V$-Signal; in gewissen Betriebsmodi muß er per Jumper deaktiviert werden (vgl. Anhang A.1). Der Spannungsregler stellt zusammen mit dem erwähnten Jumper und zwei Kondensatoren das einzige Mehr an Bauteilen dar, das aufgrund der *Low Voltage*-Entscheidung[10] erforderlich wurde.

- Versorgungsflächen: Durch Aufteilen der für die Betriebsspannung vorgesehenen Versorgungslage in zwei separate Flächen wurde verhindert, daß eine weitere Lage in Anspruch genommen werden mußte. Eine dieser Flächen implementiert das Signal $VCC$, das stets 3,3 V führt und alle Bauelemente mit Ausnahme der Taktwahl-Multiplexer (s. u.) versorgt, die andere das Signal $5V/3V$, das je nach Betriebsart 5 V oder 3,3 V führen kann (vgl. Anhang A.1).

[10]Die Taktmultiplexer arbeiten optional mit 5 V, daher handelt es sich strenggenommen um *Mixed Voltage*-Technologie.

## 3.6 Taktverteilung

Im Zusammenhang mit synchronen elektronischen Bauelementen kommt dem Taktsignal gemeinhin die größte Bedeutung zu. Es zerlegt den kontinuierlichen Lauf der Zeit in diskrete Teile und zeigt den angeschlossenen Bauelementen an, wann sie einen weiteren atomaren Operationsschritt auszuführen haben. Divergiert ein beliebiger Baustein einer synchronen Schaltung auch nur um einen Taktschritt von den übrigen, so wird er seine Umgebung i. a. grundlegend falsch interpretieren, da er sich in einem anderen internen Zustand befindet, als es die ihn umgebenden Bausteine erwarten.

### 3.6.1 Besondere elektrische Anforderungen an das Taktsignal

Sämtliche hier zum Einsatz gelangenden synchronen Bauteile reagieren auf die steigende Flanke des Taktsignals, d. h. das Überschreiten eines fest definierten elektrischen Spannungspegels wird als Auslöser für den nächsten Taktschritt betrachtet. Sie sind daher sensibel gegen elektrische Einschwingvorgänge, Leitungsreflexionen, Rauschen und andere in der Praxis unvermeidliche Phänomene, die das Risiko in sich bergen, daß der fragliche Schwellenwert unbeabsichtigt überschritten wird.

Der elektrischen Qualität des Taktsignals ist also besondere Aufmerksamkeit zu widmen, damit jede einzelne Flanke von jedem Baustein als genau eine Flanke interpretiert wird. Das vorweggeschickt, kann nun auf die in dieser Arbeit implementierte Taktverteilung eingegangen werden.

### 3.6.2 Zur Wahl stehende Taktquellen

Alle Bauelemente des Boards werden mit einem gemeinsamen Takt betrieben[11], der aus einer von drei verschiedenartigen Quellen gespeist werden kann (vgl. Abbildung 3.5):

- On-Board-Quarzoszillator: Ein Sockel für handelsübliche Quarzoszillatoren erlaubt das autarke Betreiben des Systems unabhängig von einem extern zu erzeugenden Takt und mit wechselbaren Frequenzen.
- On-Board-BNC-Buchse: Auf der Platine befindet sich eine für hochfrequente Signale geeignete BNC-Buchse, die die Versorgung mit einem extern generierten Taktsignal ermöglicht, erzeugt beispielsweise mittels eines variablen Frequenzgenerators, um die maximal erzielbare Taktfrequenz der gerade in den FPGAs implementierten Schaltung zu erproben.
- VME-seitig angelegter Takt: Die Option, den Takt aus einem der beiden Stecker *ST1* oder *ST2* zu speisen, erlaubt die Synchronisation des Prototyping-Boards mit an diesen Steckern angeschlossener externer Hardware. Das kann je nach Betriebsart ein VMEbus-Rack, ein weiteres gleichartiges FPGA-Board oder jede beliebige andere

[11] Zwar wäre es denkbar, beispielsweise jedes einzelne FPGA mit einem separaten Takt zu versorgen, etwa um eine Brücke zwischen verschieden getakteten externen Komponenten realisieren zu können, doch stünde der erwartete Nutzen in keinem gesunden Verhältnis zum erforderlichen Mehraufwand.

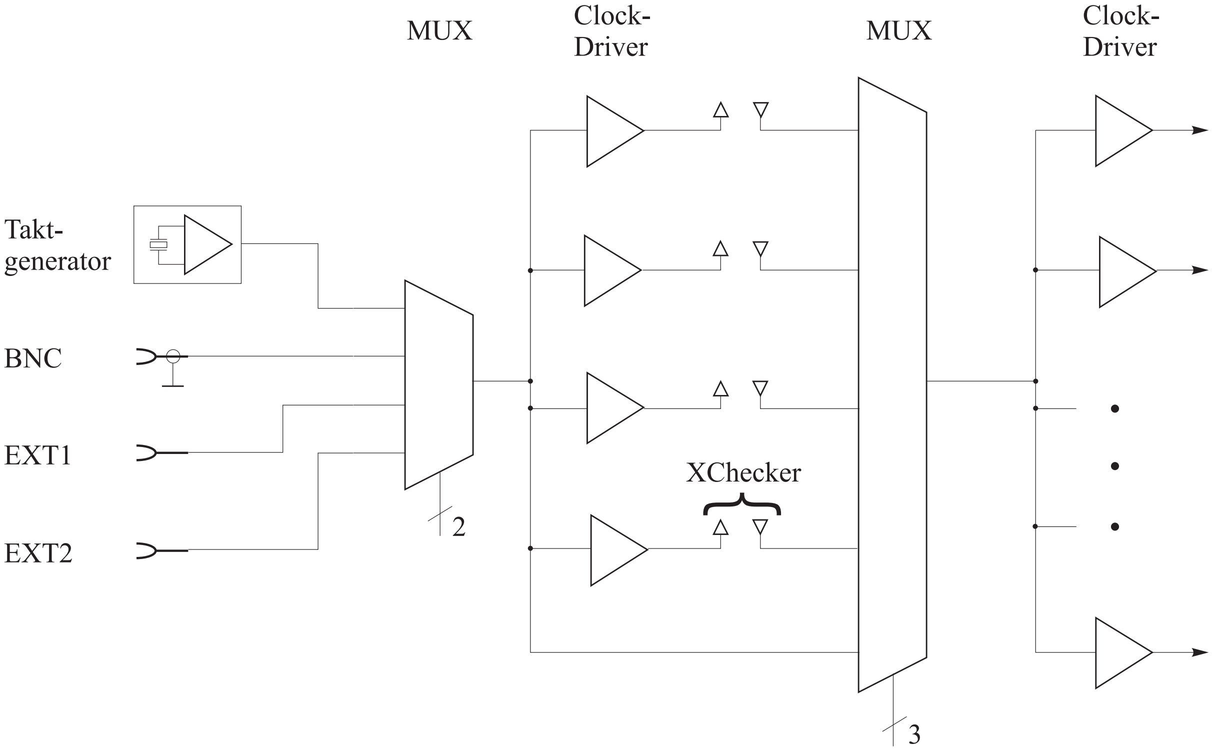


Abbildung 3.5: Taktverteilung

Hardware sein, die am betreffenden Pin ein Taktsignal liefert. Entsprechend gibt das Board den Takt am korrespondierenden Pin der Stecker *ST3* und *ST4* nach außen weiter.

### 3.6.3 Taktwahl-Multiplexer (*IC2*, *IC3*)

Wegen der oben geschilderten besonderen Ansprüche, die an die Güte des Taktsignals gestellt werden müssen, kommt es nicht in Betracht, die Auswahl der Taktquelle wie in Abbildung 3.6 (a) durch unmittelbares Verbinden über Steckbrücken (Jumper) oder einfache Schalter vorzunehmen, da diese ein erhebliches Störpotential für die hochfrequenten Signalanteile der Taktflanken darstellen. Stattdessen sind die Taktquellen mit den Eingängen des Multiplexers *IC2* verbunden, der den mittels zweier Steckbrücken identifizierten Takt durchschaltet, wie in Teil (b) der Abbildung dargestellt. Die dabei zum Einsatz kommenden Steckbrücken führen kein Taktsignal, sondern lediglich einen permanenten High- oder Low-Pegel zur Ansteuerung der Multiplexer und beeinträchtigen das Taktsignal daher nicht. (Auf die Funktion des zweiten Multiplexers *IC3* wird später im Zusammenhang mit den XChecker-Anschlüssen eingegangen.)

Als Besonderheit muß beachtet werden, daß die Eingangssignale des Multiplexers Pegel von 3,3 V oder von 5 V annehmen können; so wird der Takt beim Betrieb in einem VME64-Rack mit 5 V, beim Betrieb mit einem weiteren, identischen FPGA-Board dagegen mit dessen Betriebsspannung von 3,3 V an Stecker J1 gelegt. Folgende Alternativen wären beim Umgang mit diesem Problem denkbar:

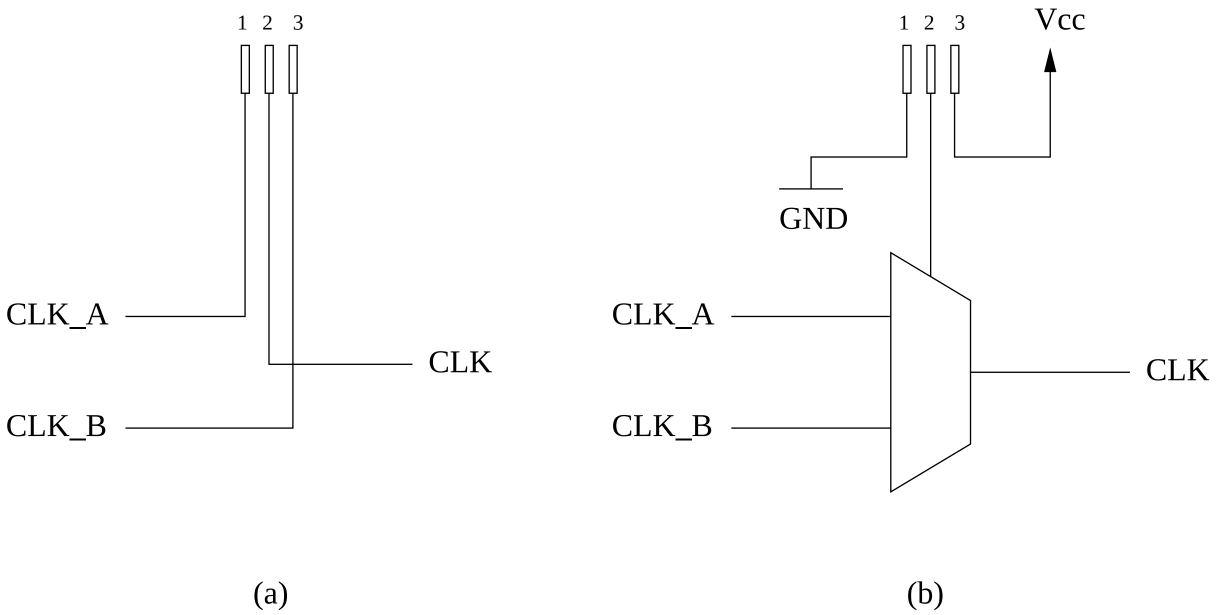


Abbildung 3.6: Auswahl einer Taktquelle: (a) mittels Jumper, (b) mittels Multiplexer

- Festlegung der Multiplexer auf eine der beiden Betriebsspannungen 5 V oder 3,3 V. Im ersten Fall könnte das Board nicht an einer 3,3-V-Spannungsquelle betrieben werden, und abgesehen davon bestünde die Gefahr, daß ein *Low Voltage*-kompatibler High-Pegel von den 5-V-Multiplexern nicht als solcher erkannt würde, daß also gälte $V_{OH,min,3V} < V_{IH,min,5V}$. Im zweiten Fall dagegen dürfte der Pegel des Taktsignals einen Wert von etwa 3,6 V nicht überschreiten; damit schieden aber sowohl der VMEbus als auch die verbreiteten Quarzoszillator-Module als Taktquelle aus.

- Es ließe sich ein dediziertes Bauelement für die Spannungswandlung des Taktes von 5 V auf 3,3 V vorschalten. Der Platzverbrauch auf der Platine sowie die zusätzliche Verzögerungszeit, die jede eingeschleifte Komponente mit sich bringt, sprechen allerdings gegen diese Option.

- Ideal wäre der Einsatz von 3,3-V-Multiplexern mit 5-V-toleranten Eingängen (analog zu den Takttreibern und FPGAs, siehe unten), doch scheinen derartige Bauelemente nicht erhältlich zu sein.

- Der in dieser Arbeit verfolgte Ansatz ist die Ausführung der Multiplexer in HC-Technologie[12]. Auf diese Weise kann *IC2* mit dem Signal $5V/3V$ gespeist werden, das in jeder Betriebsart des Boards einen geeigneten Spannungspegel zur Verfügung stellt. (Nichtsdestotrotz hat der Benutzer zu gewährleisten, daß keine Taktquelle den Pegel des $5V/3V$-Signals überschreitet; vergleiche hierzu die Beschreibung der Betriebsmodi in Anhang A.1.) Allerdings wirkt sich die vergleichsweise lange Verzögerungszeit der HC-Komponenten (z.B. 22 ns beim MM74HC251) nachteilig aus, wenn der Takt sequentiell durch mehrere FPGA-Boards geleitet werden soll.

[12] Der *High Speed CMOS*-Prozeß erlaubt Versorgungs- wie Eingangsspannungen im Bereich von 2 V bis 6 V; die in Abschnitt 3.5 genannte Bedingung, daß Eingangsspannungen maximal um etwa 0,5 V größer sein dürfen als die Versorgungsspannung, gilt aber auch hier.

### 3.6.4 Unterstützung spezieller Debugging-Features

Um die Fehlersuche beim Einsatz des Prototyping-Systems zu erleichtern, läßt sich jeder der vier XChecker-Anschlüsse in den Taktsignalweg einschalten. Sie stellen somit vier weitere potentielle Taktquellen dar, deren Auswahl mittels des zweiten Multiplexers *IC3* erfolgt, und gleichzeitig vier weitere Taktsenken, die mit den im folgenden beschriebenen Treibern angesteuert werden. Auf die Fehlersuche mittels XChecker-Anschlüssen wird in Abschnitt 3.8 näher eingegangen.

### 3.6.5 Takttreiber (*IC6*, *IC7*)

Die mittels *IC2* bzw. *IC3* ausgewählten Taktsignale müssen vervielfältigt und im ersten Fall an die Takteingänge sämtlicher synchroner Bauelemente, im zweiten Fall an die vier XChecker-Anschlüsse verteilt werden[13]. Hierbei kann kein Nutzen gezogen werden aus dem allen im CMOS-Prozeß gefertigten ICs gemeinsamen hohen *Fan Out*-Wert, aus der Tatsache also, daß jeder Ausgang eines CMOS-IC genügend Strom liefert, um zehn und mehr CMOS-Eingänge zu bedienen: würde man nämlich einen einzelnen Ausgang mehr als einen oder allenfalls zwei Takteingänge treiben lassen, so käme es wegen der Impedanzsprünge, die das Signal auf seinem Weg durch die Leiterbahn passierte, an allen Verzweigungen sowie an den Endstellen zu Reflexionen, die das ursprüngliche Signal überlagern und stören würden. Daher wurde in der vorliegenden Arbeit auf spezielle Takttreiber-Schaltkreise zurückgegriffen (*IC6* und *IC7*), die ihr Eingangssignal vervielfältigen und an mehrere Ausgänge legen. Die Impedanz jedes einzelnen Treiberausgangs (ca. 10 $\Omega$) wird mit einem geeignet dimensionierten und möglichst ortsnah angebrachten Serienwiderstand an die Leitungsimpedanz (ca. 40 $\Omega$ bei zwei parallelgeschalteten Signalsenken) angepaßt.

Wie oben beschrieben, arbeiten die den Takttreibern vorgeschalteten Taktwahl-Multiplexer unter Umständen mit 5-V-Pegeln. Daher stellt sich hier erneut die Frage, wie diese Pegel an die übrigen 3,3-V-Bauelemente angepaßt werden können; die denkbaren Optionen sind ähnlich wie die im Abschnitt „Taktwahl-Multiplexer“ genannten. Angenehmerweise zeigte sich, daß 5-V-tolerante *Low Voltage*-Takttreiber sehr wohl erhältlich sind (z.B. IDT49FCT3805 [IDT98b]), so daß hier also die ideale der vier Alternativen verwirklicht werden kann. Somit ist die Anpassung der Taktquellen-Pegel an die board-internen 3,3 V ohne jeden Mehraufwand an Bauelementen abgeschlossen.

## 3.7 Konfigurationsprozeß

In diesem Abschnitt werden die schaltungstechnischen Maßnahmen dargestellt, die im Zusammenhang mit dem Vorgang der Konfiguration der FPGAs getroffen wurden. Dieser wird soweit beschrieben, wie es für das grobe Verständnis erforderlich ist; als Referenz aber wird die Lektüre der entsprechenden Dokumentation empfohlen [Xil99] [Alf97] [Xil94].

Da die eingesetzten FPGAs ihren Inhalt bei Ausbleiben der Versorgungsspannung verlieren (vgl. Abschnitt 1.2.1), müssen sie zumindest bei jeder Inbetriebnahme konfiguriert

[13] Bei dem im Rahmen dieser Arbeit implementierten Board sind 37 Taktabnehmer zu versorgen.

werden, und auch danach kann natürlich eine Neukonfiguration erwünscht sein. Nach Anlegen der Versorgungsspannung gehen sie automatisch in einen speziellen Konfigurationszustand über. Das kann auch durch Aktivieren des $\overline{PROGRAM}$-Signals erreicht werden; weiter unten wird auf die verschiedenen Optionen eingegangen, die das Board bietet, um $\overline{PROGRAM}$ zu aktivieren und somit den Konfigurationsprozeß auszulösen.

### 3.7.1 Konfigurationsstrukturen

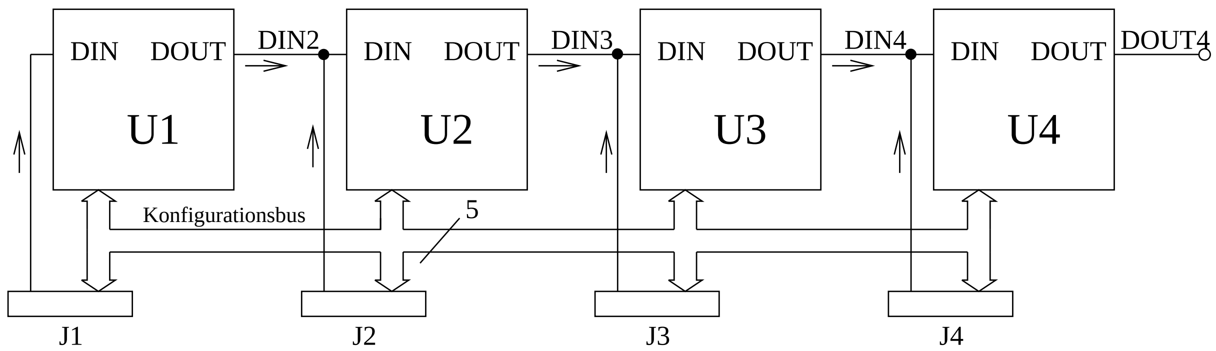


Abbildung 3.7: Schematische Darstellung der Konfigurationsstrukturen

Abbildung 3.7 zeigt die Konfigurationsstrukturen des Boards. Die wichtigste Beobachtung ist, daß die Signale $DIN2$ bis $DIN4$ die FPGAs *U1* bis *U4* zu einer Kette, einer sogenannten *Daisy Chain* verschalten, entlang derer die Konfigurationsdaten bit-seriell weitergereicht werden: vom Ausgang (*DOUT*) eines FPGAs zum Eingang (*DIN*) des nächsten.

Das erste Glied der Kette, FPGA *U1*, kann wahlweise im Slave-Modus betrieben werden (wie die übrigen FPGAs) und die Daten seines XChecker-Anschlusses *J1* entgegennehmen, oder es kann im *Master Parallel*-Modus betrieben werden und so aktiv die Konfigurationsdaten aus dem in Abschnitt 3.3.2 beschriebenen Speicher lesen. In dieser Betriebsart arbeitet das Board autark und ohne externe Datenquelle. Die Wahl des Modus von *U1* erfolgt über die Jumper *J35* bis *J37*.

Abgesehen von dieser Kette mit einem Bit Breite sind alle FPGAs an einen globalen Konfigurations-Bus angeschlossen, der aus fünf Signalen besteht. Dieses Vorgehen entspricht dem üblichen Verfahren, das der Hersteller Xilinx vorgesehen hat. Zwei Besonderheiten sind aber herauszustellen:

1. Der Konfigurations-Bus läßt sich an jedem Ort durch Jumper unterbrechen. So kann die Konfigurationskette in beliebige Abschnitte zerlegt werden, von denen jeder separat über einen XChecker-Anschluß mit Daten versorgt werden kann. Das ist beispielsweise dann von Nutzen, wenn einer der Sockel nicht mit einem FPGA, sondern mit einem Prozessormodul bestückt ist, oder wenn selektiv einige FPGAs neu programmiert werden sollen, andere aber nicht.

2. Einige Konfigurationssignale von *U1* sind auf besondere Weise beschaltet, um die dynamische Rekonfigurierbarkeit zu implementieren. Hierauf wird im folgenden näher eingegangen.

### 3.7.2 Dynamische Rekonfiguration

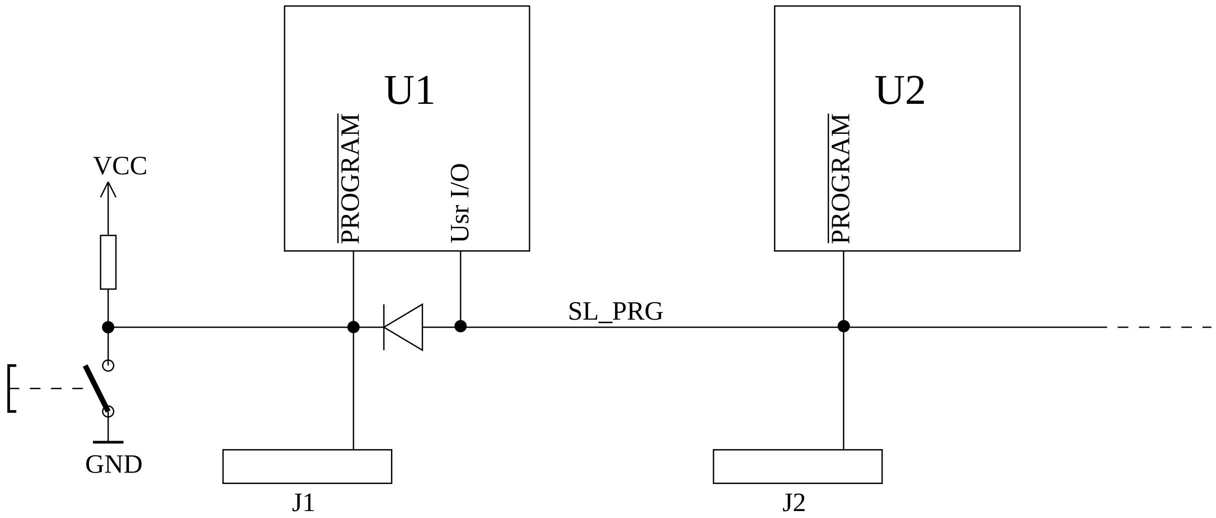


Abbildung 3.8: Slave-Program-Signal für die dynamische Rekonfiguration

Das Master-FPGA *U1* kann die übrigen FPGAs zur Laufzeit umkonfigurieren (vgl. Abschnitt 2.5). Es bleibt dabei im Normalbetrieb und unter Anwenderkontrolle, versetzt aber die übrigen FPGAs in den Konfigurationsmodus und versorgt sie je nach Ablaufphase des Gesamtsystems mit wechselnden Konfigurationsdaten, kann also während des Betriebes einige FPGAs „virtuell austauschen". Möglich wird dies durch die besondere Beschaltung zweier Konfigurationssignale (vgl. Abbildung 3.8):

- $\overline{PROGRAM}$: Üblicherweise werden die $\overline{PROGRAM}$-Pins aller FPGAs miteinander verbunden. In der vorliegenden Implementierung aber gilt das für *U1* nur in einer Richtung: Wird der $\overline{PROGRAM}$-Anschluß von *U1* auf Masse gezogen (mittels Taster *S2* oder per XChecker-Anschluß *J1*), so wirkt sich das über die Diode *D1* auf alle FPGAs aus, sie wechseln also alle in den Konfigurationsmodus; soweit stimmt die Funktionalität mit dem üblichen Vorgehen überein. Wenn dagegen das gemeinsame Programmiersignal der nachgeschalteten FPGAs — im folgenden $\overline{SL\_PRG}$ für *Slave Program* genannt — auf Masse gezogen wird, so verhindert die Diode, daß auch *U1* neu programmiert wird. Über ein benutzerprogrammierbares *U1*-Pin hat der Anwender Zugriff auf eben dieses $\overline{SL\_PRG}$-Signal. Es ist ihm also möglich, den durch die oben erwähnten Jumper eingestellten und bei *U2* beginnenden Abschnitt der Konfigurationskette (mindestens *U2* allein, maximal *U2* bis *U4*) in den Konfigurationsmodus zu versetzen, während die übrigen FPGAs unter Benutzerkontrolle bleiben.

- $CCLK$: Nachdem *U1* mittels des Signals $\overline{SL\_PRG}$ die Rekonfiguration angestoßen hat, muß es an seinem $CCLK$-Pin den Konfigurationstakt ausgeben. $CCLK$ ist aber genau wie $\overline{PROGRAM}$ nicht benutzerprogrammierbar. Deswegen ist analog zu $\overline{SL\_PRG}$ ein Signal $SL\_CCLK$ (*Slave CCLK*) vorgesehen, das der Benutzer über ein frei programmierbares FPGA-Pin treiben kann. Über einen Jumper[14] läßt es sich mit dem *CCLK*-Bus verbinden; es treibt dann zwar auch den *CCLK*-Eingang von *U1* selbst, doch das hat keine Auswirkungen.

Als Quelle für die Konfigurationsdaten kann das Master-FPGA *U1* den Konfigurationsspeicher oder auch den *Single Port*-Arbeitsspeicher benutzen, denn die dafür erforderlichen Signale *D*0 bis *D*7, *A*0 bis *A*19, *HDC* und $\overline{LDC}$ sind benutzerprogrammierbar. Auch das *DOUT*-Signal von *U1* ist benutzerprogrammierbar. Daher ist es dem Anwender ohne weitere Vorkehrungen möglich, die (in *U1* zu serialisierenden) Konfigurationsdaten am Pin *DOUT* auszugeben; die nachgeschalteten FPGAs bemerken keinen Unterschied zur herkömmlichen Konfiguration.

### 3.7.3 Konfigurieren eines Verbundes aus mehreren FPGA-Boards

Wie in Abschnitt 3.4 beschrieben, lassen sich viele FPGA-Boards miteinander verschalten, so daß ein großer Verbund entsteht, der eine fast beliebige Topologie aufweisen kann. Drei Alternativen sind denkbar, um ein solches System zu konfigurieren:

1. Jedes Board einzeln, mit separatem *Start Up*: Es spricht nichts dagegen, die Tatsache zu ignorieren, daß mehrere Boards im Verbund betrieben werden und ganz konventionell jedes Board einzeln zu konfigurieren — sei es aus dem lokalen Konfigurationsspeicher, oder sei es über einen oder mehrere XChecker-Anschlüsse. Die FPGAs beenden ihre *Start Up*-Sequenz innerhalb jedes Boards synchron, Board-übergreifend jedoch zu verschiedenen Zeitpunkten. Mit geringem Aufwand können die FPGAs so programmiert werden, daß ein solcher Verbund die Arbeit aufnimmt, sobald das letzte Board konfiguriert ist.

2. Jedes Board einzeln, aber mit gemeinsamem *Start Up*: Wie im vorigen Fall hat jedes Board seine eigene Konfigurationsdatenquelle, aber die Signale *DONE* und $\overline{INIT}$ werden Board-übergreifend zu einem globalen Bus zusammengeschaltet. Zu diesem Zweck können sie an beliebigen XChecker-Anschlüssen abgegriffen werden. Als Resultat beenden alle FPGAs global im gleichen Takt den *Start Up*-Prozeß, so daß dem Benutzer die Aufgabe abgenommen wird, auf das letzte Board zu warten. Es muß im Einzelfall geprüft werden, wie viele Boards sich derartig verschalten lassen, denn die vier genannten Konfigurationssignale sind auf jedem einzelnen Board mit Pull-Up-Widerständen verbunden, die im Verbund parallel geschaltet wirken. Werden beispielsweise fünf Boards mit Pull-Up-Widerständen von 20 kΩ[15] zusammengeschaltet, so stellt ihre Parallelschaltung einen Widerstand von 20 kΩ / 5 = 4 kΩ dar. Wenn also im ungünstigsten Fall ein einzelnes FPGA-Pin ein solches Signal auf Masse halten muß, so fließt ein Strom von 3,3 V / 4 kΩ < 1 mA, der weit unterhalb der zulässigen Grenze von 12 mA bleibt.

[14]Dieser Jumper *JP39* ist nicht zwingend erforderlich; er kann immer gesetzt bleiben, doch wurde er vorgesehen, um die Handhabung des Systems bei Verzicht auf die dynamische Rekonfigurierbarkeit der üblichen Handhabung anzupassen.

[15]Xilinx empfiehlt 10 kΩ bis 50 kΩ.

3. Mehrere Boards gemeinsam: Die Notwendigkeit, jedes einzelne Board des Verbundes mit Konfigurationsdaten zu versorgen, läßt sich eliminieren, indem die oben beschriebene *Daisy Chain*, also die Kette seriell weitergereichter Konfigurationsdaten, auf jeweils nachfolgende Boards ausgedehnt wird. Dafür kann der Datenausgang des letzten Kettengliedes innerhalb jedes Boards, also das $DOUT$-Signal von FPGA *U4*, an einem separaten Anschlußpin *JP38* abgegriffen und an den ersten XChecker-Anschluß *J1* eines anderen Boards weitergereicht werden. Die Signale $DONE$ und $\overline{INIT}$ des Konfigurations-Busses sind wie im vorigen Fall auf alle involvierten Boards auszuweiten. Auch $CCLK$ muß nun auf die gleiche Weise verteilt werden.

Mischformen sind natürlich realisierbar. So können beispielsweise drei Boards aus dem Verbund eine autarke Konfigurationskette bilden, die aus dem Konfigurationsspeicher des ersten dieser Boards gespeist wird, während weitere Boards einzeln über XChecker-Anschlüsse programmiert werden.

## 3.8 Unterstützung bei der Fehlersuche

### 3.8.1 Readback

Alle drei in Abschnitt 2.6.1 beschriebenen Varianten für das Auslesen der FPGA-Zustandsinformationen werden durch geeignete Beschaltung aller XChecker-Anschlüsse unterstützt, ohne irgendwelche benutzerprogrammierbaren I/O-Pins für diesen Zweck zu opfern. (Die Signale $RT$ und $RD$ wurden ressourcensparend an zwei der Konfigurationsmodus-Pins angeschlossen, und als Readback-Takt wird das ohnehin nicht benutzerprogrammierbare $CCLK$-Pin benutzt.) Jedes der vier FPGAs kann daher mit Methoden analog zu Breakpoints und Single-Stepping aus der Softwarewelt auf Entwurfsfehler untersucht werden.

### 3.8.2 Boundary Scan

Auch die *Boundary Scan*-Fähigkeiten der FPGAs werden voll unterstützt. Der Ausgang $TDO4$ der Scan-Kette kann an einem gesonderten Pin des letzten FPGAs *U4* abgegriffen werden. Als Kopf der Kette kann wahlweise *U1* oder *U2* eingesetzt werden. So ist es möglich, das Boundary-Scan-Feature selbst dann zu nutzen, wenn im *U1*-Sockel ein dazu inkompatibles Prozessormodul installiert ist.

## 3.9 Test und Validierung

Das beschriebene Prototyping-System wurde mit CAD-Software umgesetzt. Automatische Verfahren verifizierten die Übereinstimmung von Platinenlayout und Schaltplan. Die Platine wurde beim Hersteller einem E-Test unterzogen, der die vollständige Übereinstimmung des gefertigten Produkts mit dem Layout sicherstellt; die Platine ist somit geprüft und einsatzbereit.

# Kapitel 4

# Zusammenfassung und Ausblick

Im Rahmen dieser Diplomarbeit wurde ein FPGA-Board für den Anwendungsschwerpunkt des *Rapid Prototyping* entwickelt, das sich aufgrund seiner Universalität gleichermaßen auch für andere Aufgabenstellungen eignet. Es kann mit seinem *On Board*-Speicher sinnvoll autark eingesetzt werden. Durch den Einsatz von Dual-Port-Speicher sowie geteiltem Single-Port-Speicher können verschiedenste Kommunikationsstrategien umgesetzt werden. Nichtflüchtiger Konfigurationsspeicher eliminiert die Nachteile SRAM-basierter FPGAs; er kann optional auch als Arbeitsspeicher eingesetzt werden und ist zu jedem Zeitpunkt von außen zugreifbar.

Durch sein skalierbares Modulkonzept erlaubt das Board die Realisierung großer, aus vielen einzelnen Komponenten bestehender Gesamtsysteme mit unterschiedlichsten Topologien. Unter Zuhilfenahme einfacher Bus-Platinen sind auch dreidimensionale Strukturen realisierbar. Die Größe solcher Systeme ist aufgrund der Abwesenheit jeglicher zentraler Komponenten unbegrenzt. Dank der VMEbus-Konformität kann auf eine breite Palette bestehender Hardware-Lösungen zurückgegriffen werden.

Die Option, ein FPGA durch ein DSP- oder Mikroprozessor-Modul zu ersetzen, erlaubt die gemeinsame Integration von Hardware- und Software-Komponenten auf einem einzelnen Prototypenboard, was vielfältige Einsatzmöglichkeiten eröffnet.

Das Board unterstützt sowohl die aktive als auch die passive Konfiguration sowie sämtliche Debugging-Hilfsmittel der FPGAs. Dies gestattet es, Hardware-Entwürfe mit Methoden ähnlich Breakpoints und Single-Step-Modus zu untersuchen.

# Anhang A

# Bedienungshinweise

Viele für die Bedienung des hier realisierten Boards relevante Aspekte werden in Kapitel 3 behandelt, da sie sehr eng mit der technischen Realisierung des Systems verknüpft sind. Ein Verständnis des genannten Kapitels ist für den Betrieb der Platine ohnehin unerläßlich. Noch fehlende Angaben werden an dieser Stelle nachgereicht.

## A.1 Spannungsversorgung

Das Prototyping-Board kann wahlweise mit 3,3 V oder mit 5 V betrieben werden; im zweiten Fall wandelt Spannungsregler *IC5* die angelegten 5 V um in 3,3 V. Die Spannung kann über die Anschlüsse *ST1*, *ST2*, *X1* oder Kombinationen aus diesen zugeführt werden. Die Versorgung innerhalb des Boards geschieht über die beiden Signale $5V/3V$ sowie $VCC$; aus dem Schaltplan in Abbildung B.6 geht hervor, wie diese Signale mit den genannten Anschlüssen, dem Spannungsregler und dem Jumper *JP32* verschaltet sind. Daraus ergibt sich, daß während des Betriebs folgende Bedingungen erfüllt sein müssen:

1. $VCC$ muß 3,3 V führen, entweder durch direktes Anlegen dieser Spannung an *X1-3* oder durch Einsatz des Spannungsreglers.

2. Das Signal $5V/3V$ muß 5 V oder 3,3 V führen; es ist durch einen der Anschlüsse *ST1*, *ST2* oder *X1-1* oder Kombinationen aus diesen zu versorgen. Im VMEbus-Betrieb wird es über *ST1* und *ST2* vom VMEbus gespeist.

3. Wird $VCC$ aus *X1-3* gespeist, so ist dieses Signal duch Öffnen von Jumper *JP32* vom Ausgang des Spannungsreglers zu trennen.

4. Bei Verwendung des Spannungsreglers muß das Signal $5V/3V$ einen Pegel von 5 V führen. Ferner muß Jumper *JP32* gesetzt sein. *VCC* darf dann nicht aus *X1-3* gespeist werden, kann aber dort abgegriffen werden, um weitere Boards zu versorgen, falls dies angebracht erscheint. Der Strom durch den Spannungsregler darf 7 A nicht übersteigen.

5. Steht keine 5-V-Quelle zur Verfügung, etwa weil das Board in einer 3,3-V-Backplane arbeitet, so kann es auch mit einer einzelnen 3,3-V-Quelle betrieben werden. Der Spannungsregler ist durch Öffnen von *JP32* zu deaktivieren. Das Signal $VCC$ kann zweckmäßigerweise mittels einer Drahtbrücke von *X1-1* an *X1-3* gelegt werden.

6. Da die Taktwahlmultiplexer mit dem $5V/3V$-Signal gespeist werden, darf sein Pegel den der eingesetzten Taktquelle nicht unterschreiten. Wird der Takt durch einen XChecker-Anschluß geleitet, so darf das $5V/3V$-Signal aus dem gleichen Grund auch dessen Pegel nicht unterschreiten.

In jedem Fall steht das $5V/3V$-Signal an den Steckern *ST3* und *ST4* eventuell nachgeschalteten FPGA-Boards als Spannungsquelle zur Verfügung.

## A.2 VMEbus-Interface-Konfiguration

Das Prototyping-Board eignet sich für den Einsatz im VMEbus gleichermaßen wie für den Betrieb im direkten Verbund mit weiteren Boards. Die VMEbus-Stecker *ST1* und *ST2* müssen hierfür an die jeweilige Betriebsart angepaßt werden; das geschieht mittels einiger Jumper, die zwischen den genannten Interfaces und den zugehörigen FPGAs positioniert sind. Die Tabellen A.1 und A.2 enthalten eine Auflistung der erforderlichen Konfigurationen; die Jumper sind in der Reihenfolge ihres Auftretens auf der Platine aufgeführt. Für dreipolige Jumper ist mit „12“ oder „23“ angegeben, welche der beiden möglichen Pin-Paare kurzzuschließen sind, bei zweipoligen Jumpern steht „o“ für offen, „x“ für gesetzt. Die Angabe $Usr$ kennzeichnet benutzerdefinierbare Signale.

| | VME64-Modus | | Verbund-Modus | |
|---|---|---|---|---|
| | Position | Signal | Position | Signal |
| *JP22* | 12 | $\overline{BCLR}$ | 23 | $GND$ |
| *JP9* | 23 | $GND$ | 12 | $Usr$ |
| *JP18* | 23 | $GND$ | 12 | $Usr$ |
| *JP20* | 23 | $GND$ | 12 | $Usr$ |
| *JP28* | 23 | $GND$ | 12 | $Usr$ |
| *JP27* | 23 | $GND$ | 12 | $Usr$ |
| *JP19* | o | ./. | x | $GND$ |

Tabelle A.1: Jumper-Plazierung für *ST1*

## A.3 Auswahl der Taktquelle

Bei der Beschreibung der Taktverteilung in Abschnitt 3.6 wurde auf die detaillierte Angabe der erforderlichen Jumper-Konfigurationen für die Taktquellenwahl verzichtet; sie wird hiermit nachgereicht (vgl. Abbildungen B.1 und 3.5).

| | VME64-Modus | | Verbund-Modus | |
|---|---|---|---|---|
| | Position | Signal | Position | Signal |
| *JP23* | o | ./. | x | $Usr$ |
| *JP24* | 12 | $GND$ | 23 | $Usr$ |
| *JP25* | o | ./. | x | $Usr$ |
| *JP26* | 23 | $GND$ | 12 | $Usr$ |

Tabelle A.2: Jumper-Plazierung für *ST2*

Tabelle A.3 behandelt die Auswahl einer der vier möglichen Taktquellen mittels Multiplexer *IC2*. Soll der vom VMEbus zur Verfügung gestellte Takt von 16 MHz benutzt werden, so ist als Taktquelle *ST1* zu wählen.

| Gewünschte Taktquelle | *JP15* | *JP14* |
|---|---|---|
| Taktgenerator *QG1* | 23 | 23 |
| BNC-Buchse *J5* | 23 | 12 |
| Interface *ST1* | 12 | 23 |
| Interface *ST2* | 12 | 12 |

Tabelle A.3: Jumper-Plazierung für *IC2*

Für Debugging-Zwecke kann der Systemtakt durch einen der vier XChecker-Anschlüsse geleitet werden. Der zweite Taktwahl-Multiplexer *IC3* wird hierfür gemäß Tabelle A.4 angesteuert.

| Gewünschte Taktquelle | *JP17* | *JP16* | *JP30* |
|---|---|---|---|
| Original-Signal von *IC2* | 23 | 23 | 23 |
| XChecker-Anschluß *J1* | 12 | 23 | 23 |
| XChecker-Anschluß *J2* | 12 | 23 | 12 |
| XChecker-Anschluß *J3* | 12 | 12 | 23 |
| XChecker-Anschluß *J4* | 12 | 12 | 12 |

Tabelle A.4: Jumper-Plazierung für *IC3*

## A.4 Auswahl des Betriebsmodus von *U1*

Das FPGA *U1* als erstes Glied der Konfigurationskette kann optional im Master-Modus betrieben werden, um sich und die übrigen FPGAs selbständig zu konfigurieren. Die Auswahl des Modus geschieht mittels dreier Jumper gemäß Tabelle A.5 [Xil99].

| Gewünschte Betriebsart | *JP35* | *JP36* | *JP37* |
|---|---|---|---|
| Slave Serial | 23 | 23 | 23 |
| Master Parallel Up | 12 | 12 | 23 |
| Master Parallel Down | 12 | 23 | 23 |

Tabelle A.5: Jumper-Plazierung für *U1*

## A.5 Segmentieren des Konfigurationsbusses

Falls nicht alle FPGAs des Boards benötigt werden, oder wenn nur bestimmte FPGAs dynamisch rekonfiguriert werden sollen, so kann der Konfigurationsbus an geeigneten Stellen aufgetrennt werden, wie in Tabelle A.6 dargestellt (vgl. Abbildung B.2 sowie [Xil99]). Spalte *U2–U3*, Zeile $DONE$ beispielsweise benennt denjenigen Jumper, der das Signal $DONE$ zwischen den FPGAs *U2* und *U3* verbindet bzw. trennt.

| Signal | *U1–U2* | *U2–U3* | *U3–U4* |
|---|---|---|---|
| $CCLK$ | *JP1* | *JP2* | *JP6* |
| $DONE$ | *JP10* | *JP5* | *JP12* |
| $\overline{INIT}$ | *JP7* | *JP8* | *JP29* |
| $\overline{PROGRAM}$ | *JP4* | *JP3* | *JP11* |

Tabelle A.6: Jumper zum Auftrennen des Konfigurationsbusses

## A.6 Bestückungsdruck

In Abbildung A.1 ist der Bestückungsdruck des hier implementierten FPGA-Boards wiedergegeben. Er ist leicht verkleinert abgebildet; die Originalgröße der Platine beträgt 233,35 mm x 160,00 mm (ohne überstehende Stecker).

## A.7 Sonstiges

Über die Pfostenleiste *JP21* kann von außen auf den Konfigurationsspeicher zugegriffen werden. Zwei der Pins von *JP21* sind mit benutzerprogrammierbaren Pins der FPGAs *U1* und *U2* verbunden (vgl. Abbildung B.3). So ist es möglich, *JP21* während des Betriebs der FPGAs als Speicher-Interface beispielsweise für einen externen DSP einzusetzen. Eine der beiden Leitungen könnte als $REQUEST$-Signal eingesetzt werden, um vom DSP den Zugriff auf den Speicher zu erbitten, die andere wäre dann zweckmäßigerweise mit einem vom DSP generierten $GRANT$-Signal beschaltet.

Bei Einsatz der dynamischen Rekonfigurierung ist es hilfreich, wenn das Master-FPGA *U1* den Zustand des $DONE$-Signals lesen kann. Zu diesem Zweck kann $DONE$ an einem der XChecker-Anschlüsse abgegriffen und mit einem Kabel an einen der FPGA-Eingänge geführt werden, beispielsweise an Pin 1 oder 2 von *JP21*.

Die beiden LEDs *D2* und *D4* erfüllen keine besondere Aufgabe und dienen dem Anwender als Status-Anzeige zur Verfügung.

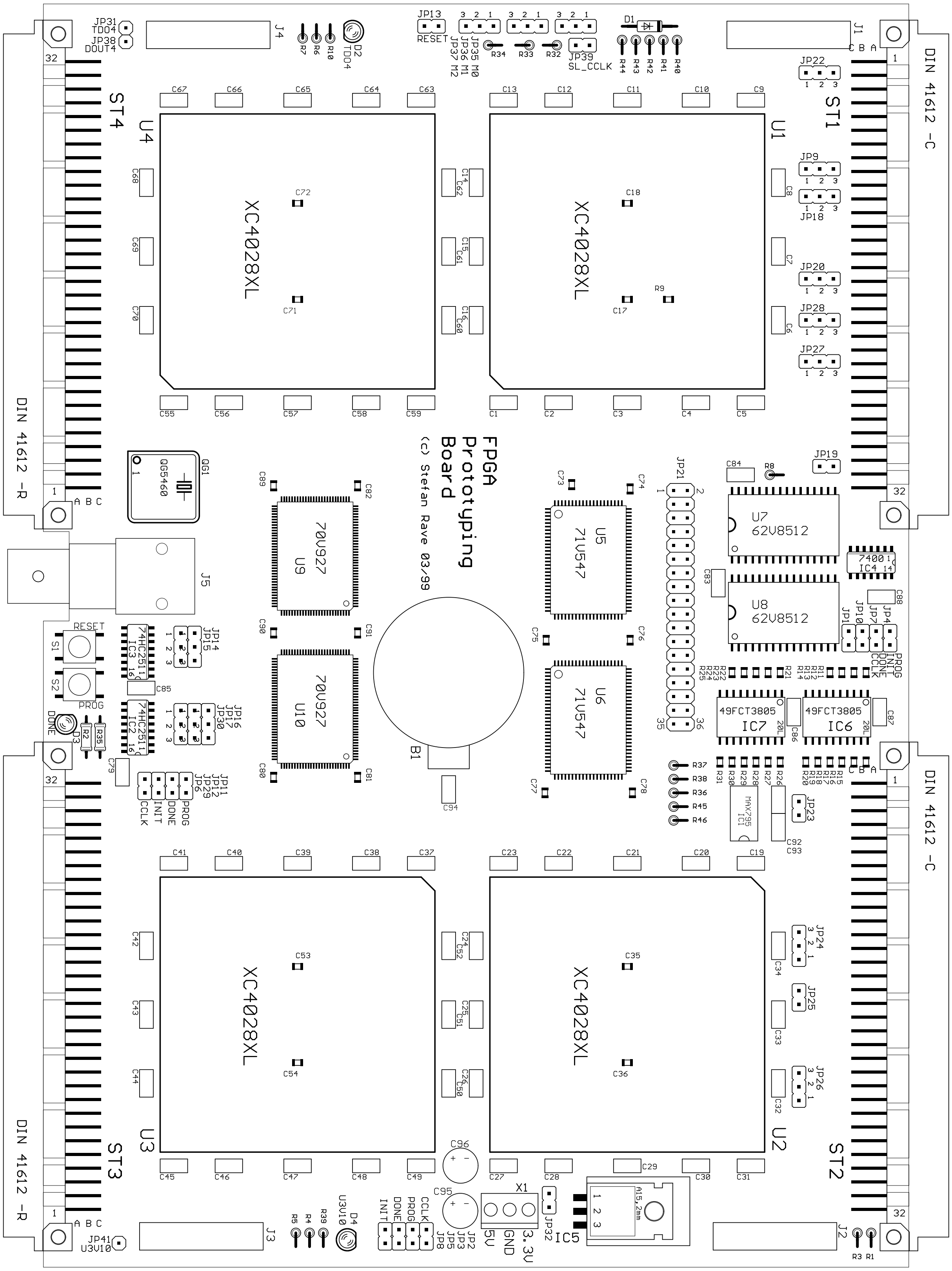


Abbildung A.1: Bestückungsdruck der FPGA-Platine (leicht verkleinert)

# Anhang B

# Schaltplan

Auf den folgenden Seiten ist der vollständige Schaltplan des FPGA-Boards dargestellt. Er ist vorzugsweise im Querformat zu lesen. Komplexe Bauteile sind in mehrere einzelne Schaltungssymbole zerlegt, die wiederum auf verschiedene Blätter verteilt sein können; so findet sich beispielsweise der Nordteil des FPGAs *U1* in Abbildung B.10 bezeichnet mit den Namen *U1N1* bis *U1N5*, während seine Konfigurations-Pins in Abbildung B.2 als *U1CFG* dargestellt sind.

Auch viele der abgebildeten Netze und Busse sind auf mehrere Blätter verteilt. Erscheint also auf verschiedenen Seiten der gleiche Netzname, so handelt es sich um dasselbe elektrische Signal. Zum Verständnis des Schaltplanes ist daher die in Anhang C abgedruckte Netzliste unentbehrlich: So ist beispielsweise auf der ersten Seite des Schaltplanes in Abbildung B.1 am linken Rand (im Querformat betrachtet) das Signal $CLK2$ zu sehen, dessen aus dem Blatt weisender Pfeil andeutet, daß es auf mindestens einem anderen Blatt fortgeführt wird[1]. Ein Blick in die alphabetisch sortierte Netzliste zeigt, daß $CLK2$ auch auf den Blättern 7 und 12 auftritt, in beiden Fällen verbunden mit dem FPGA *U2*.

[1] Busse werden nie mit einem solchen Pfeil gekennzeichnet.

## Schaltplan: Übersicht

| Blatt | Inhalt |
|---|---|
| 1 | Taktverteilung (Quellen, Multiplexer und Treiber) |
| 2 | Konfigurationssignale, XChecker-Anschlüsse |
| 3 | Konfigurationsspeicher, Batterie-Pufferung |
| 4 | Arbeitsspeicher *Single Port* |
| 5 | Arbeitsspeicher *Dual Port* |
| 6 | Interface *ST1*, *U1E*, *U1S*, Stromversorgung |
| 7 | Interface *ST2*, *U2N*, *U2E* |
| 8 | Interface *ST3*, *U3E*, *U3S* |
| 9 | Interface *ST4*, *U4N* |
| 10 | *U1N*, *U4S* |
| 11 | *U1W*, *U2W* |
| 12 | *U2S*, *U3N* |
| 13 | *U3W*, *U4W* |
| 14 | *U4E* |
| 15 | Entkopplungskondensatoren |

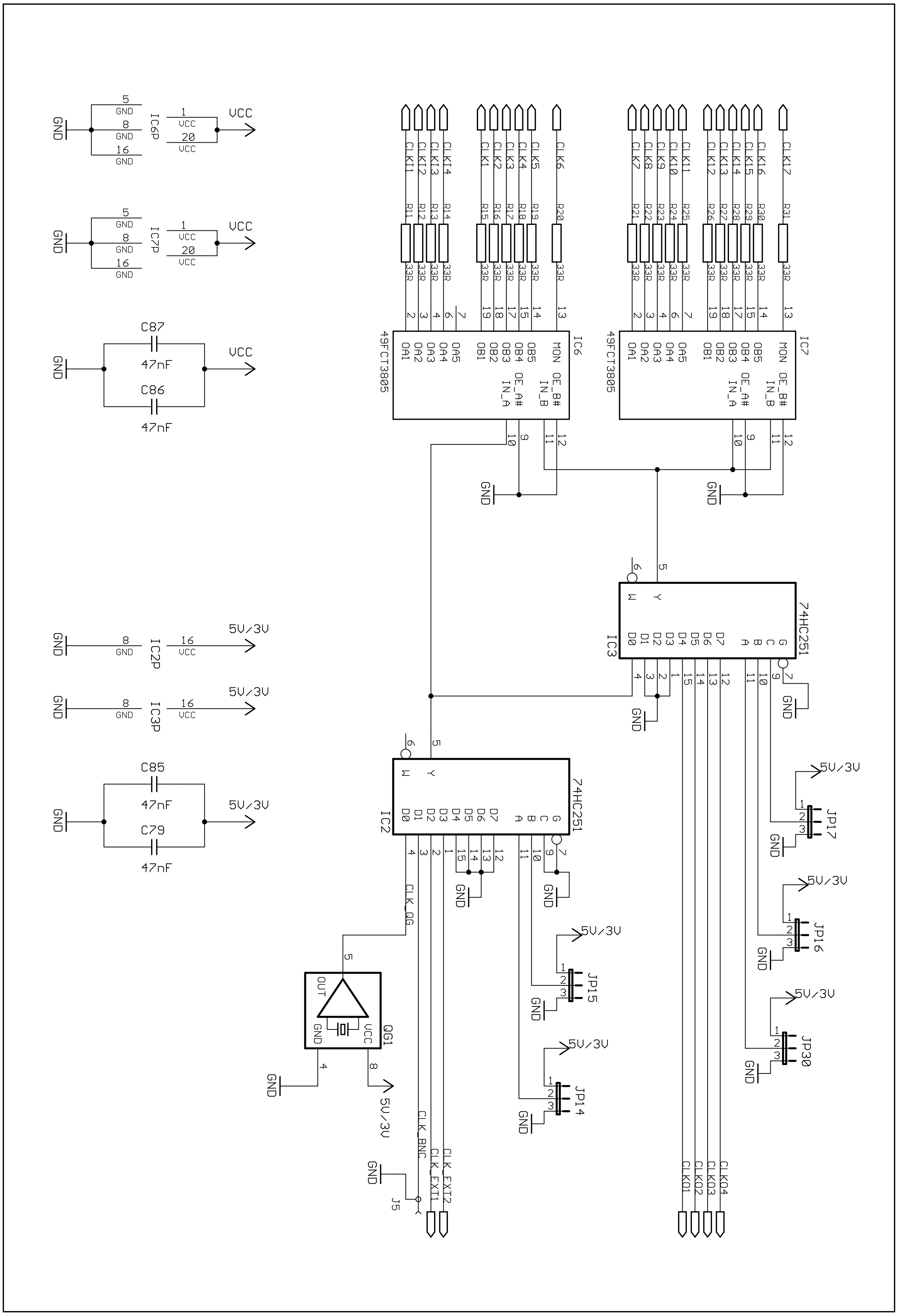


Abbildung B.1: Taktverteilung (Quellen, Multiplexer, Treiber)

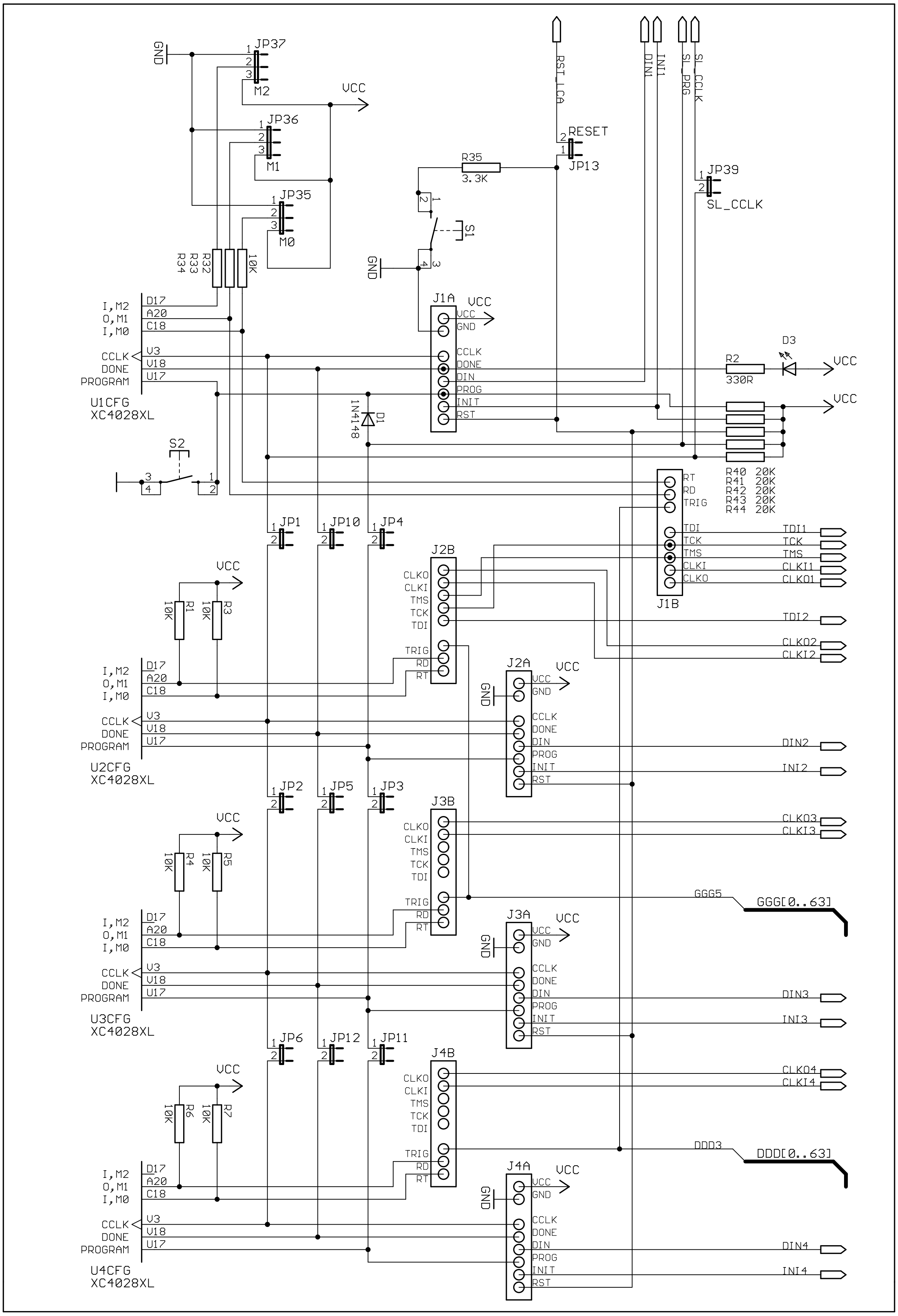


Abbildung B.2: Konfigurationssignale, XChecker-Anschlüsse

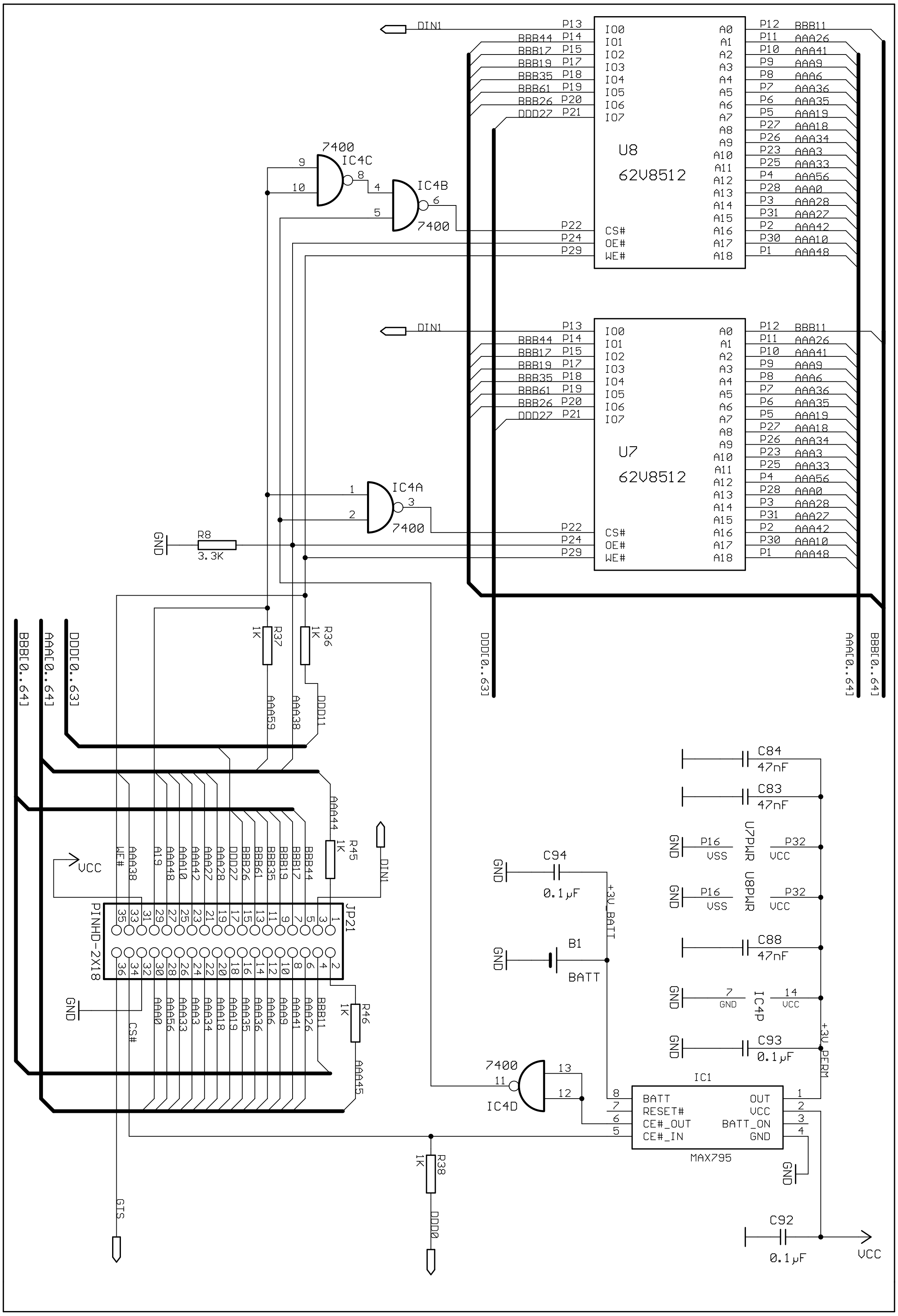


Abbildung B.3: Konfigurationsspeicher mit Batteriepufferung

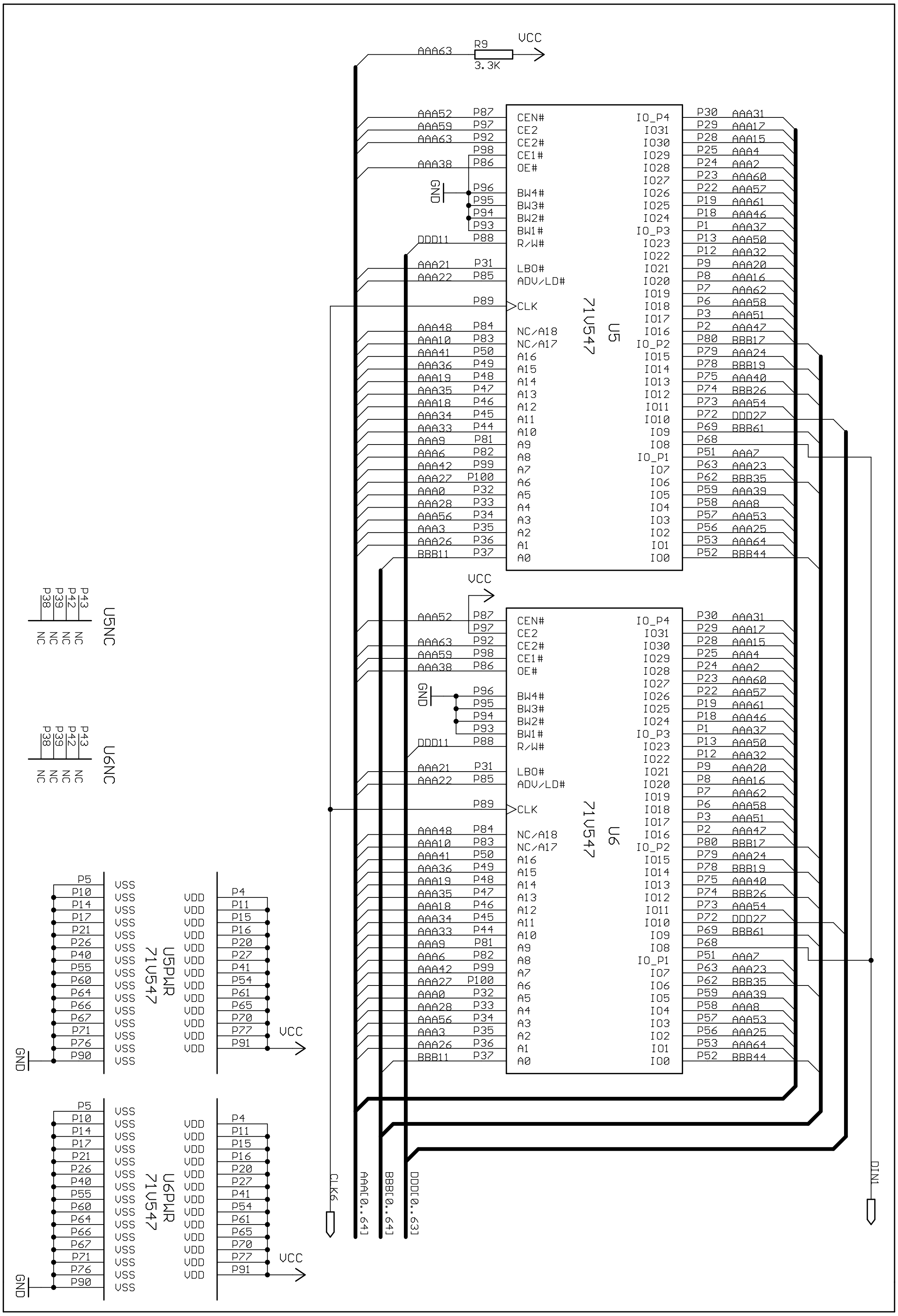

Abbildung B.4: Arbeitsspeicher *Single Port*

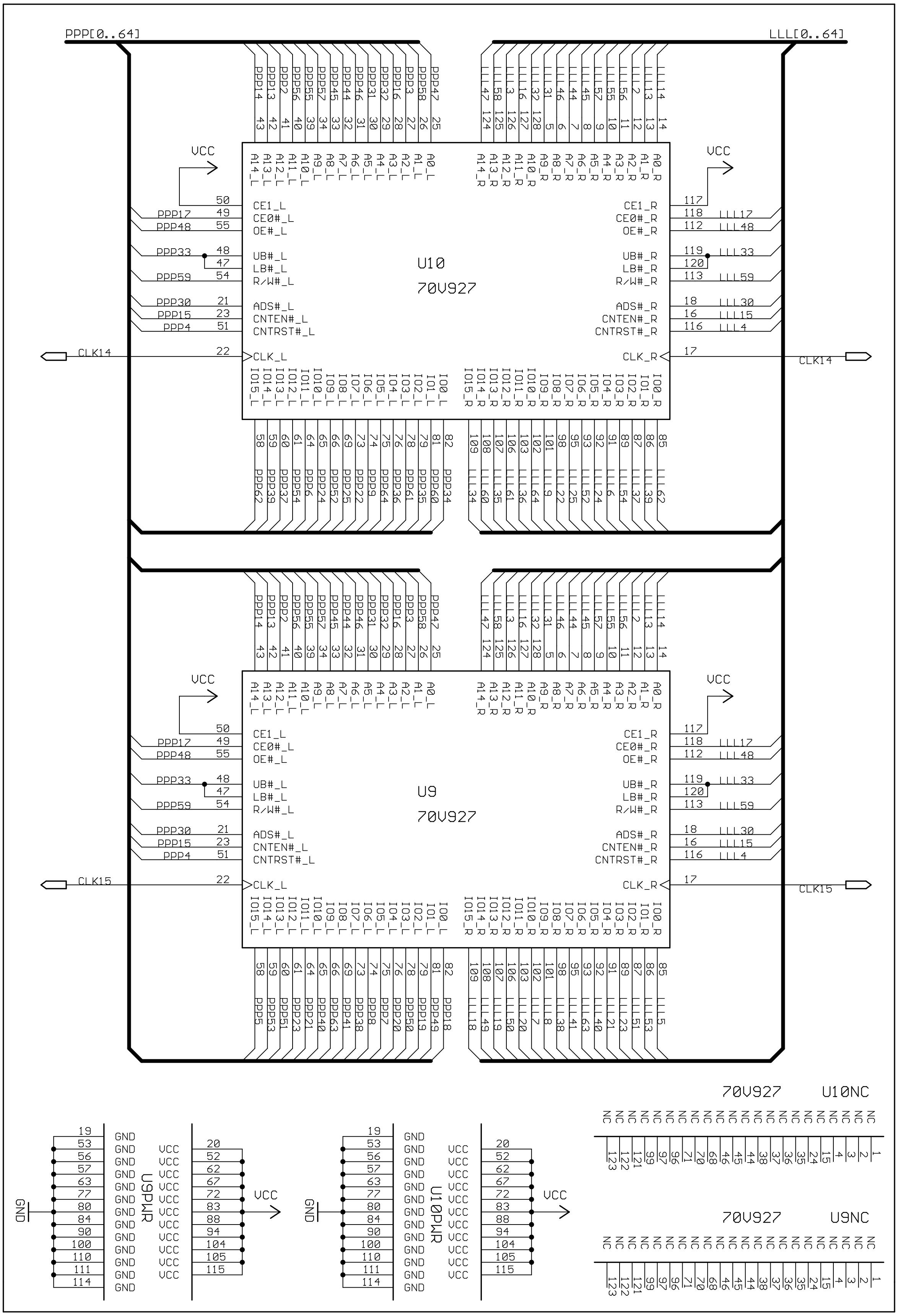


Abbildung B.5: Arbeitsspeicher *Dual Port*

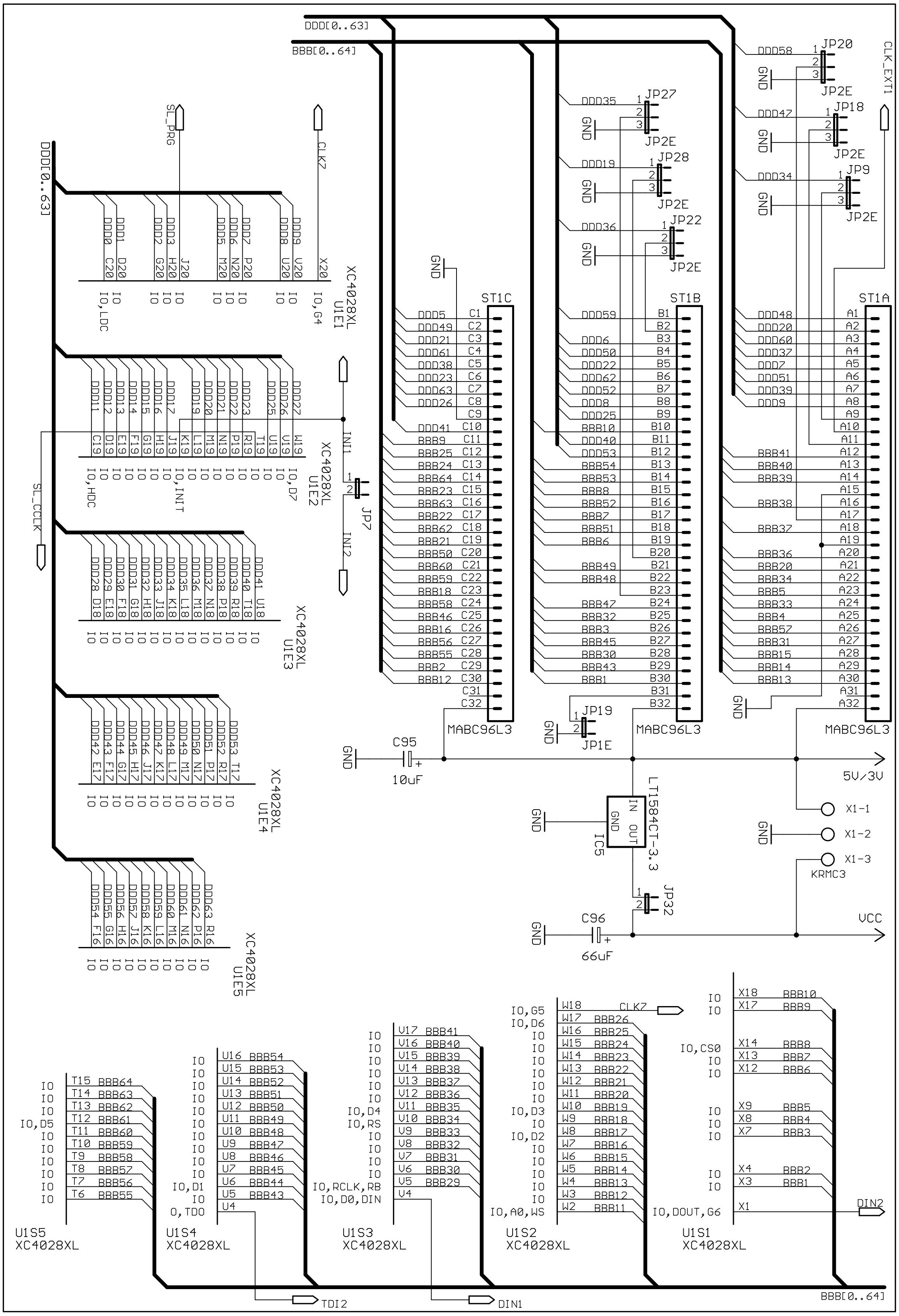


Abbildung B.6: Interface *ST1*, *U1*, Stromversorgung

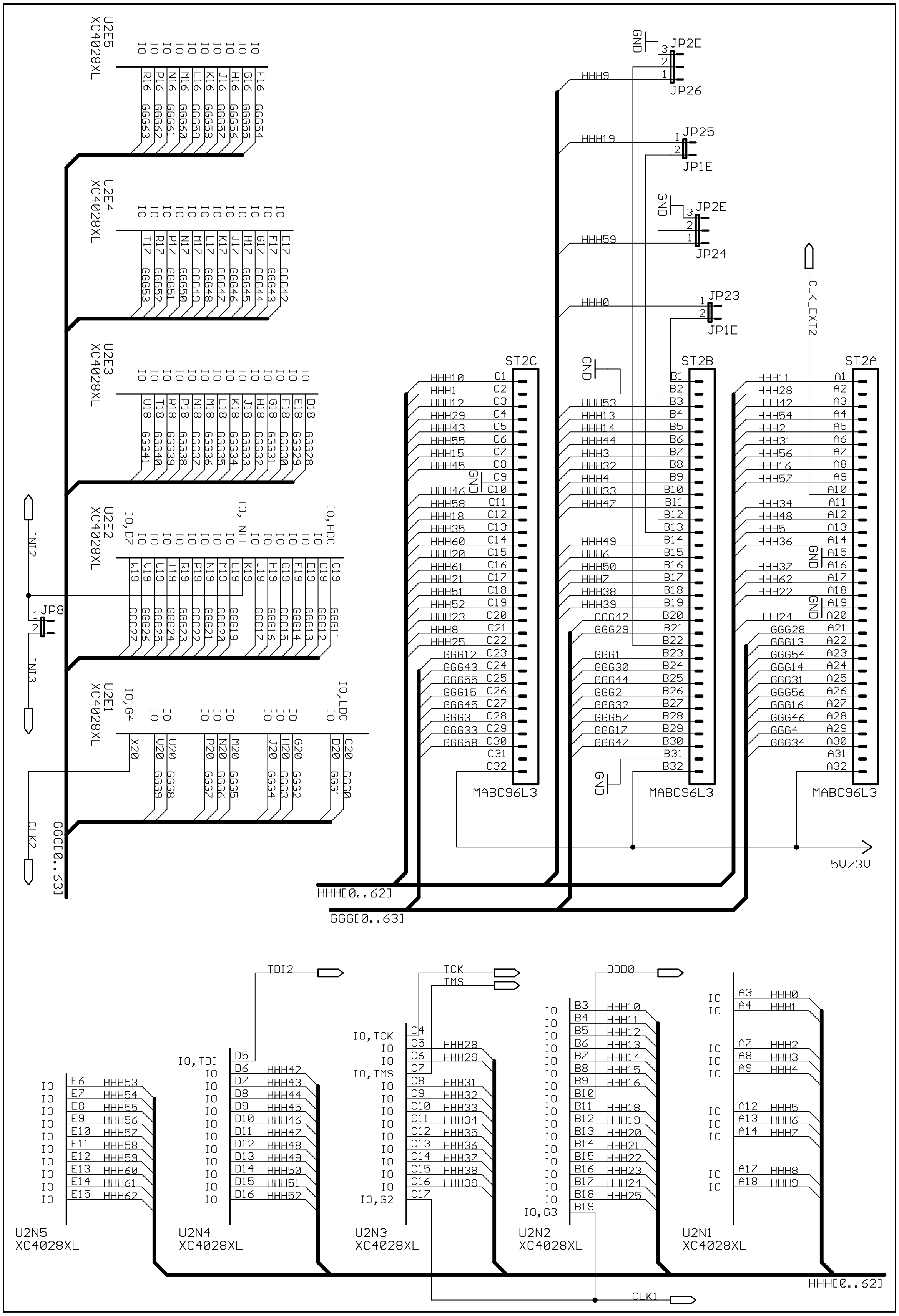

Abbildung B.7: Interface *ST2*, *U2*

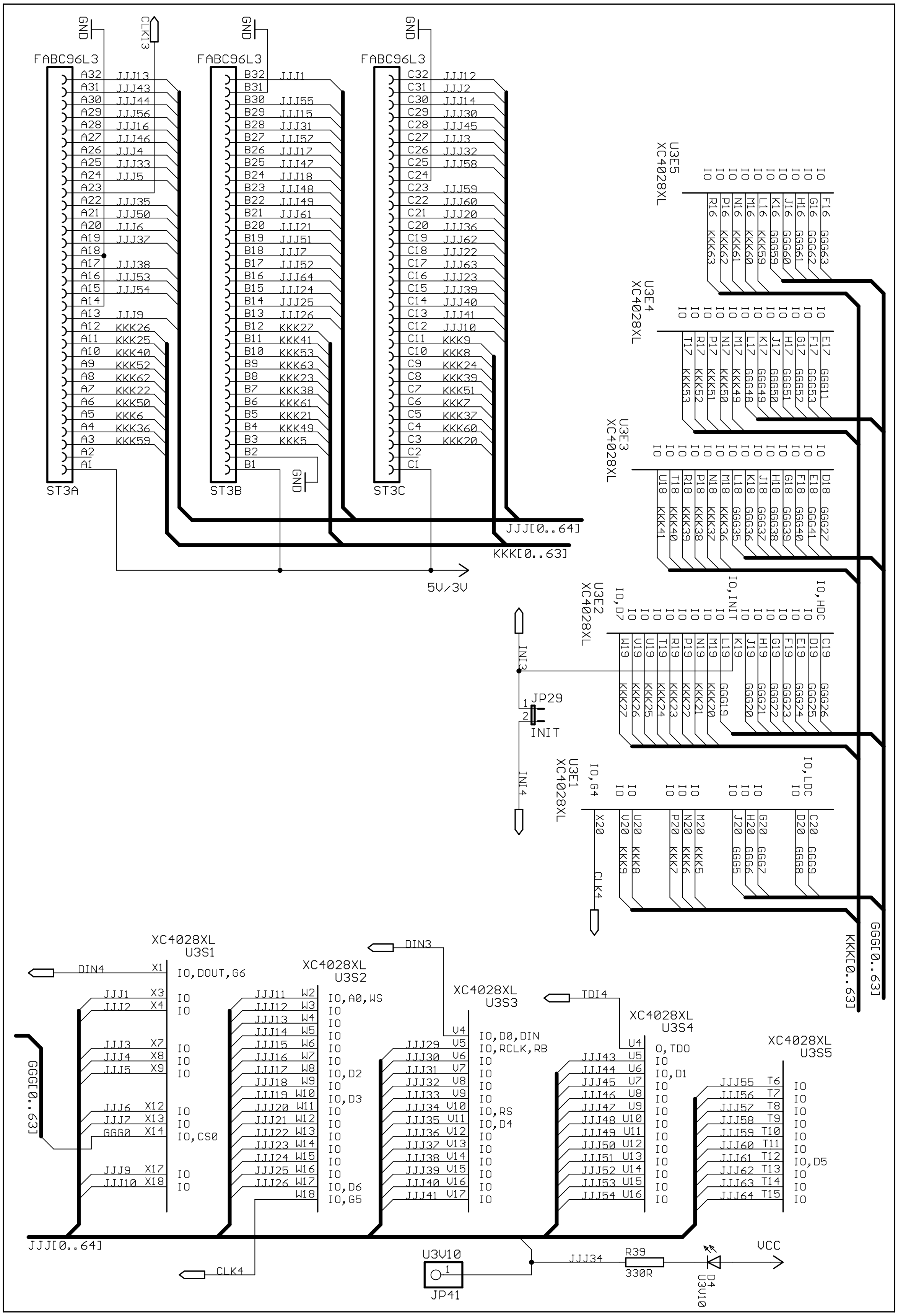


Abbildung B.8: Interface *ST3*, *U3*

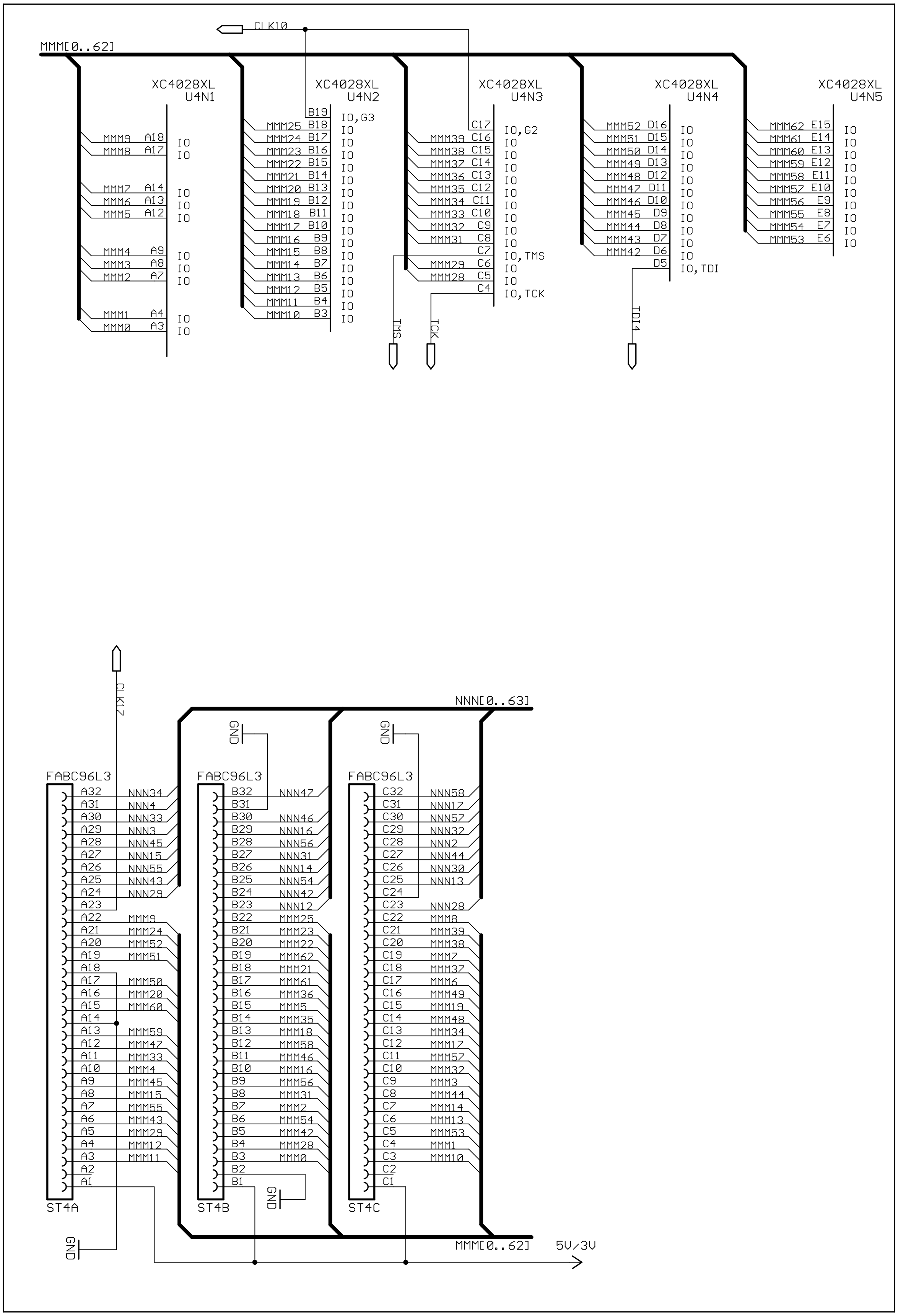


Abbildung B.9: Interface *ST4*, *U4*

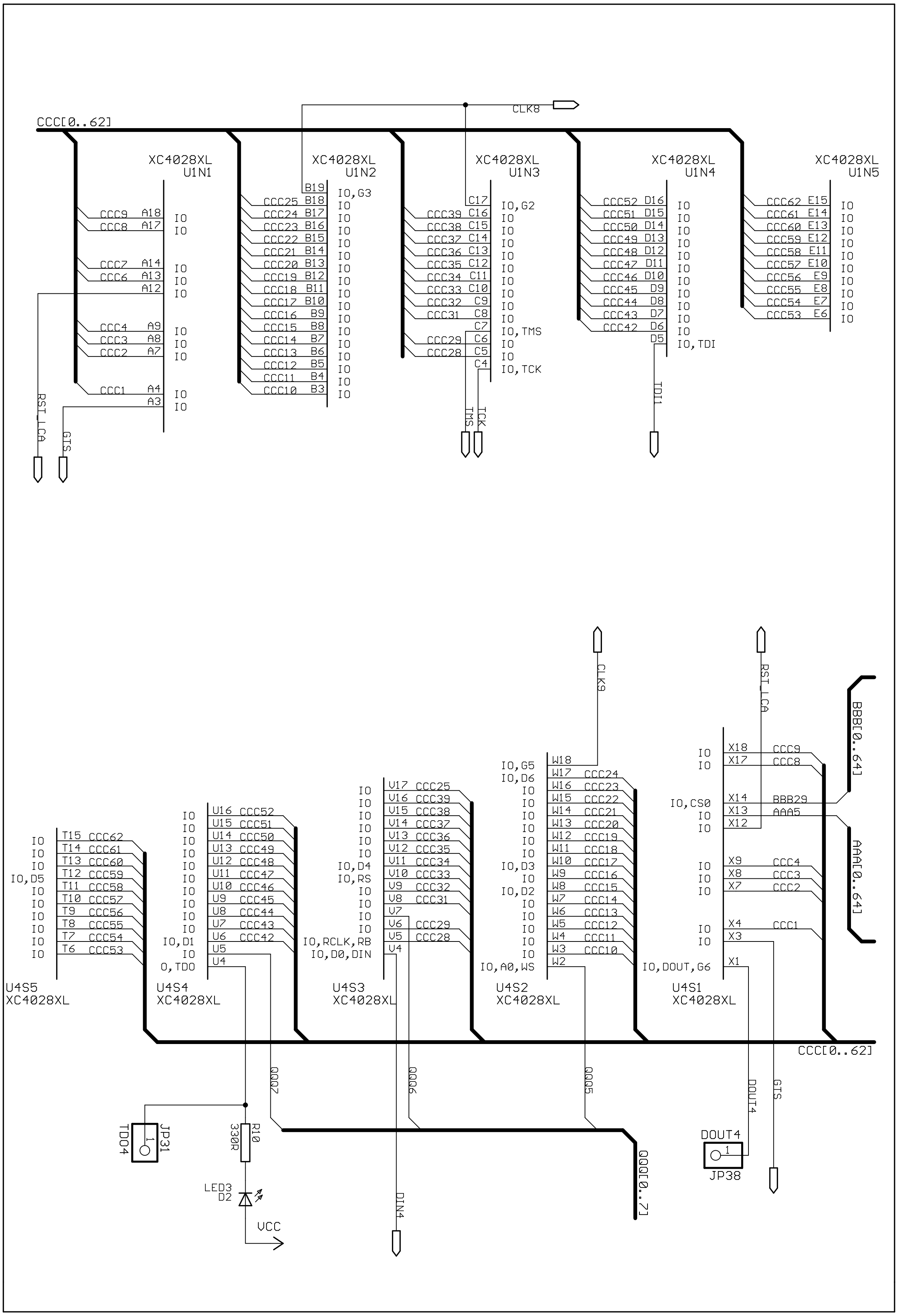


Abbildung B.10: *U1* und *U4*

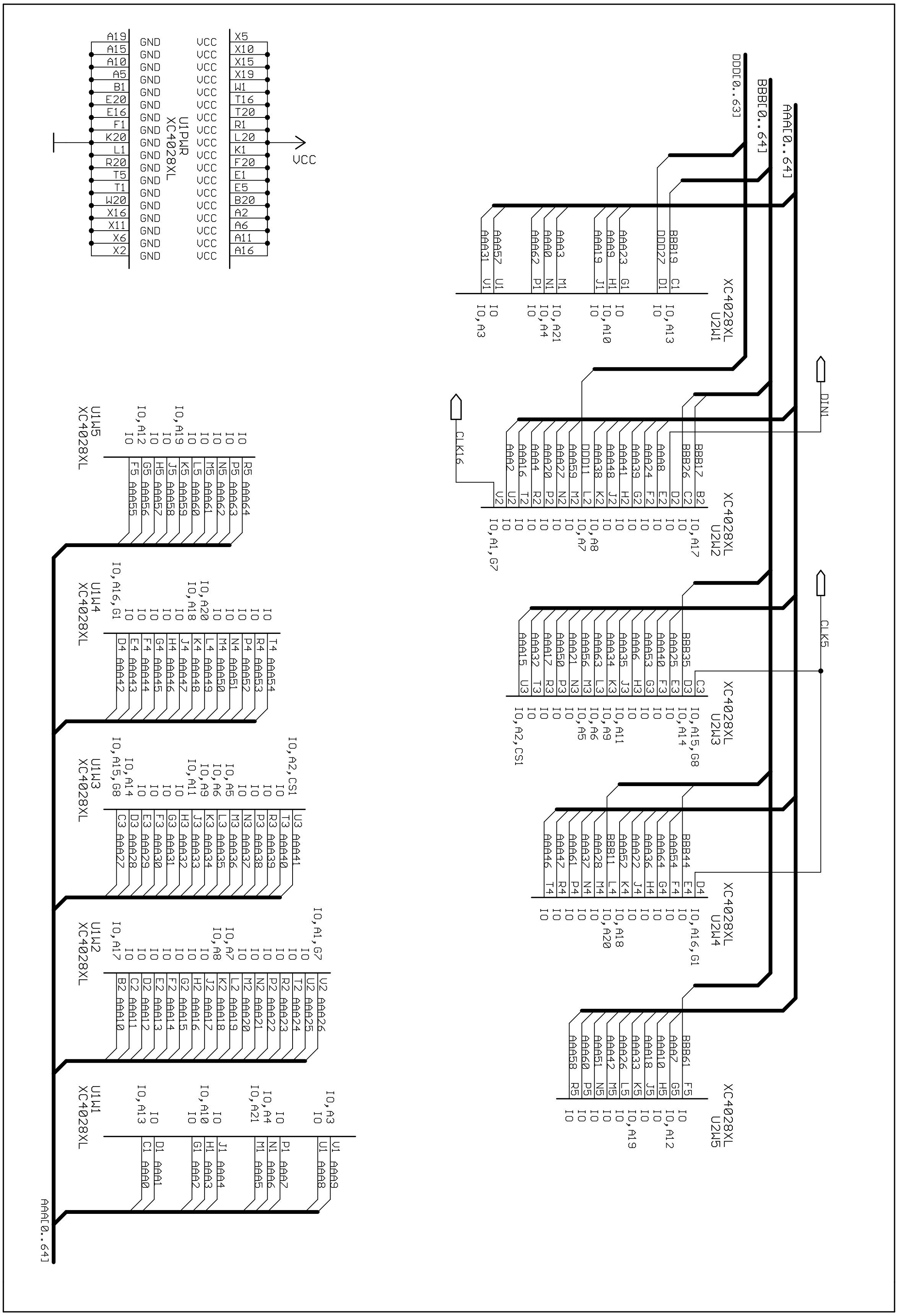

Abbildung B.11: *U1* und *U2*; Bus *AAA*: gemeinsames *Single Port RAM*

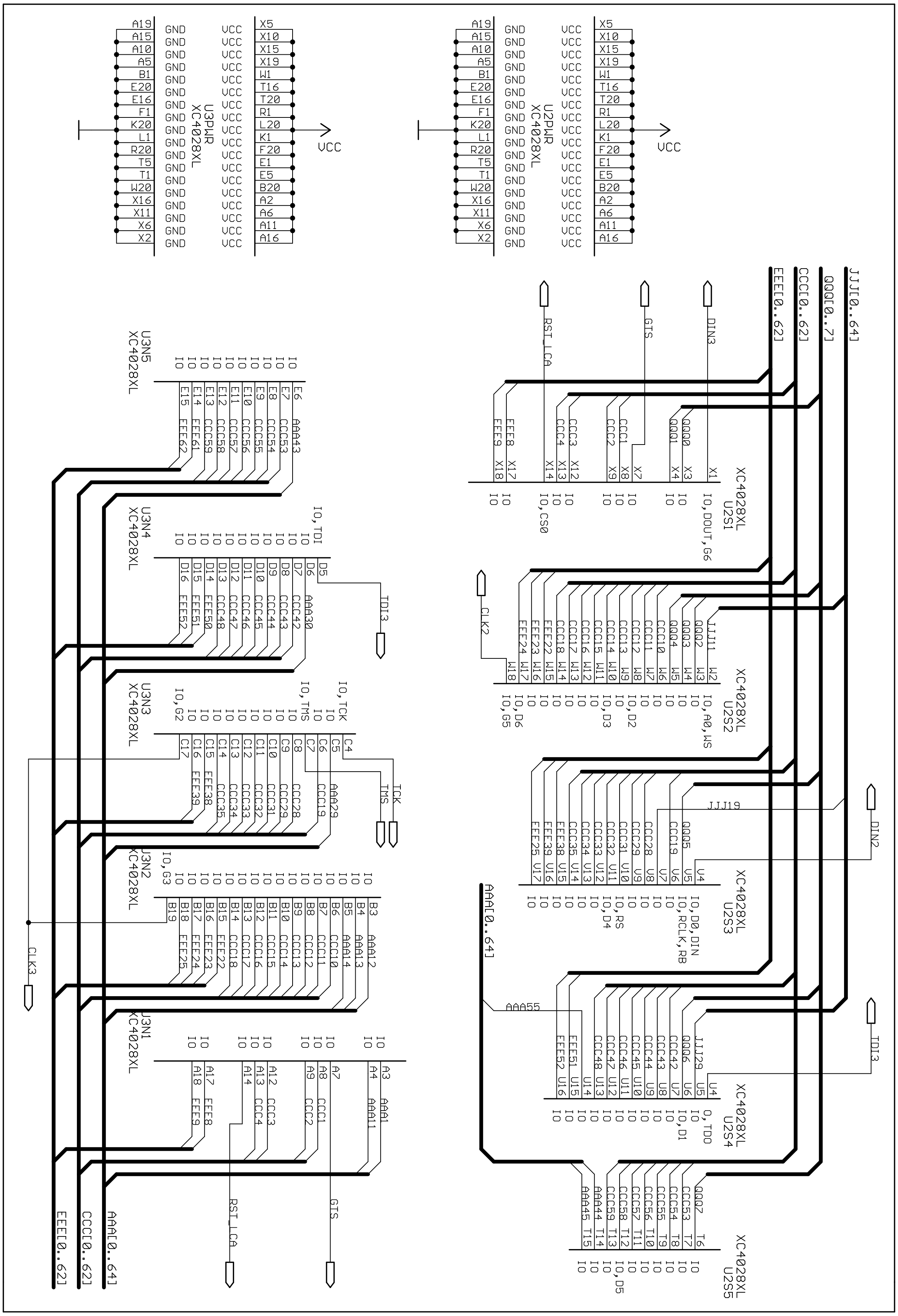

Abbildung B.12: *U2* und *U3*

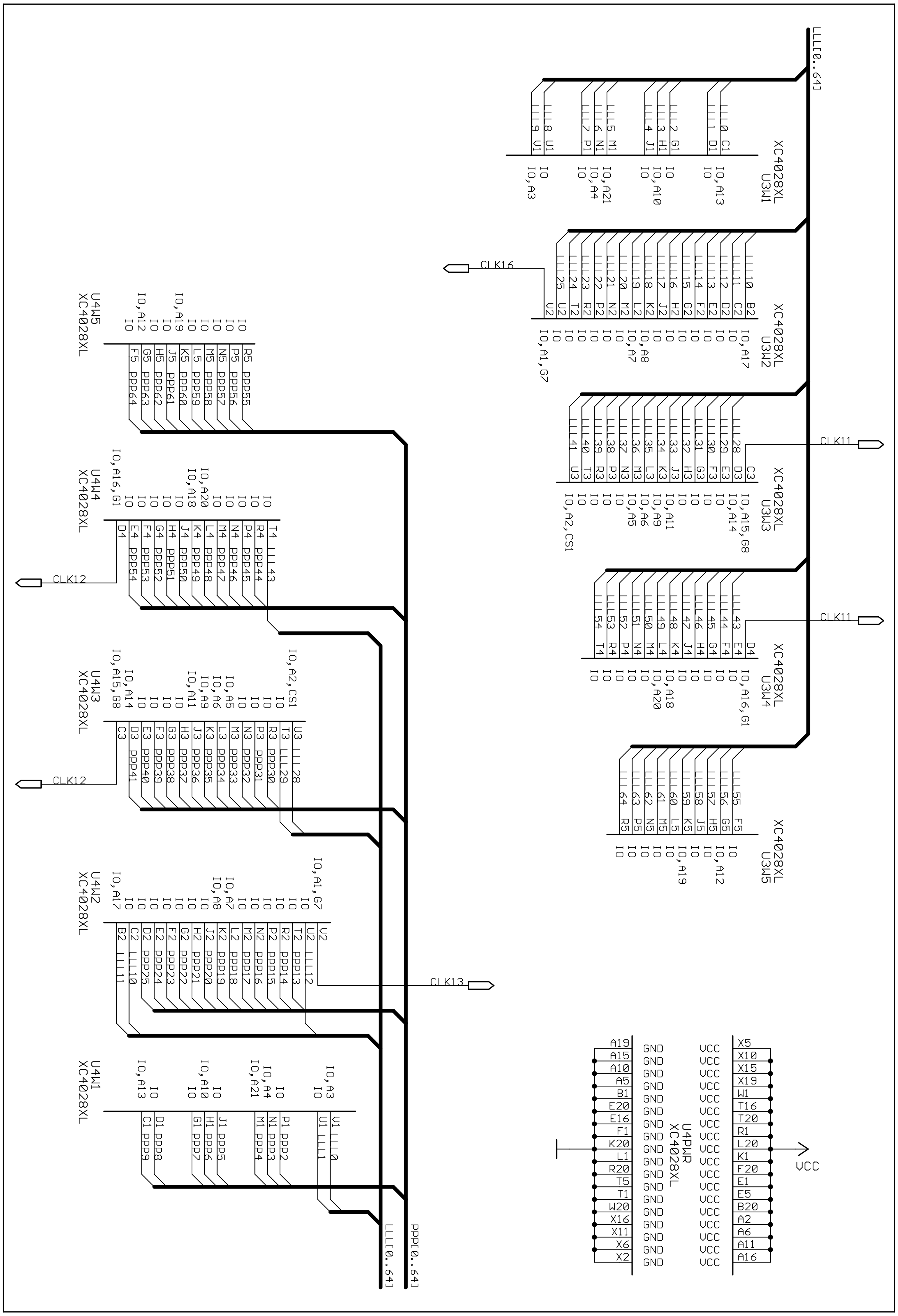


Abbildung B.13: *U3* und *U4*; Bus *PPP*: gemeinsames *Dual Port RAM*

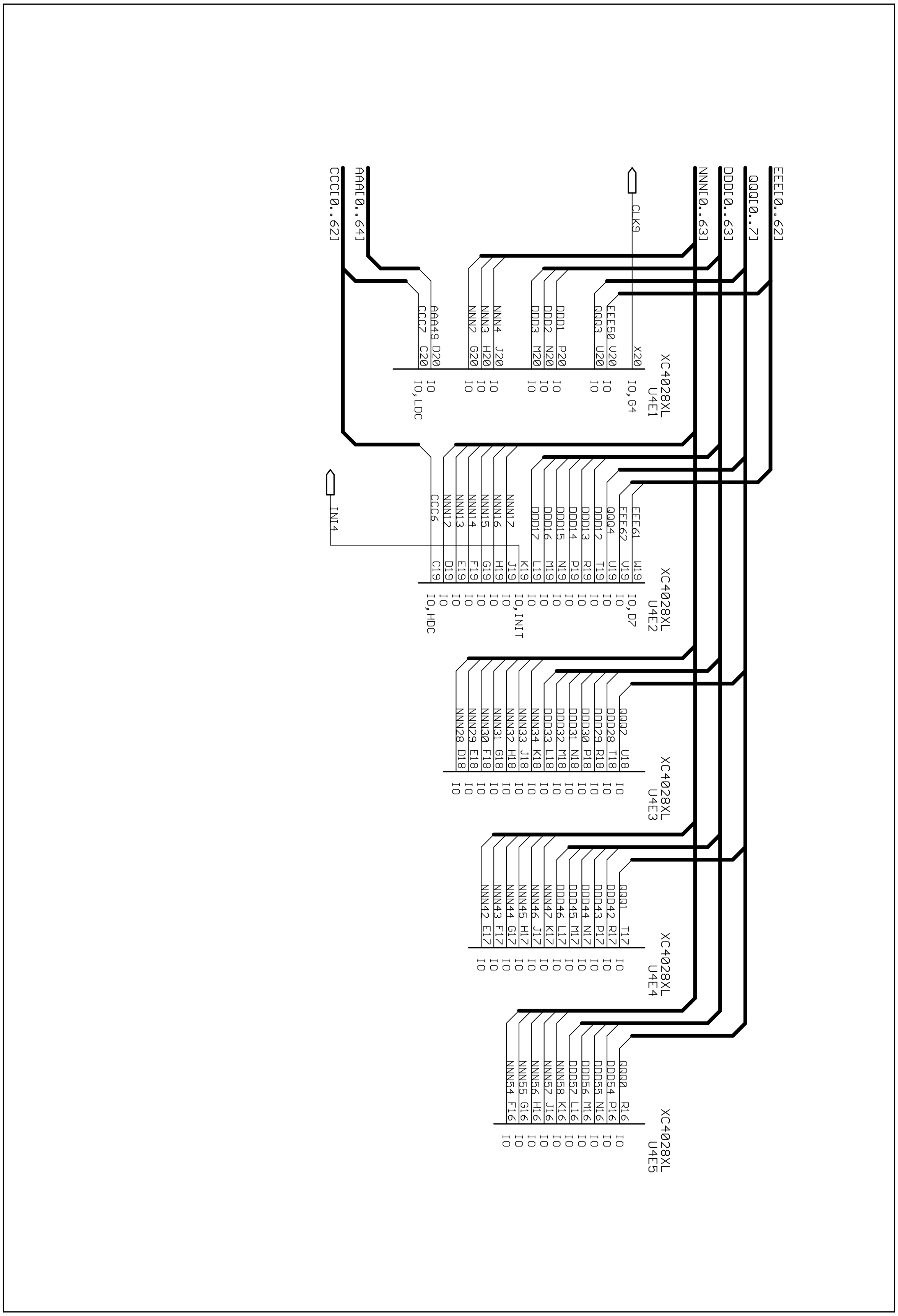


Abbildung B.14: *U4*

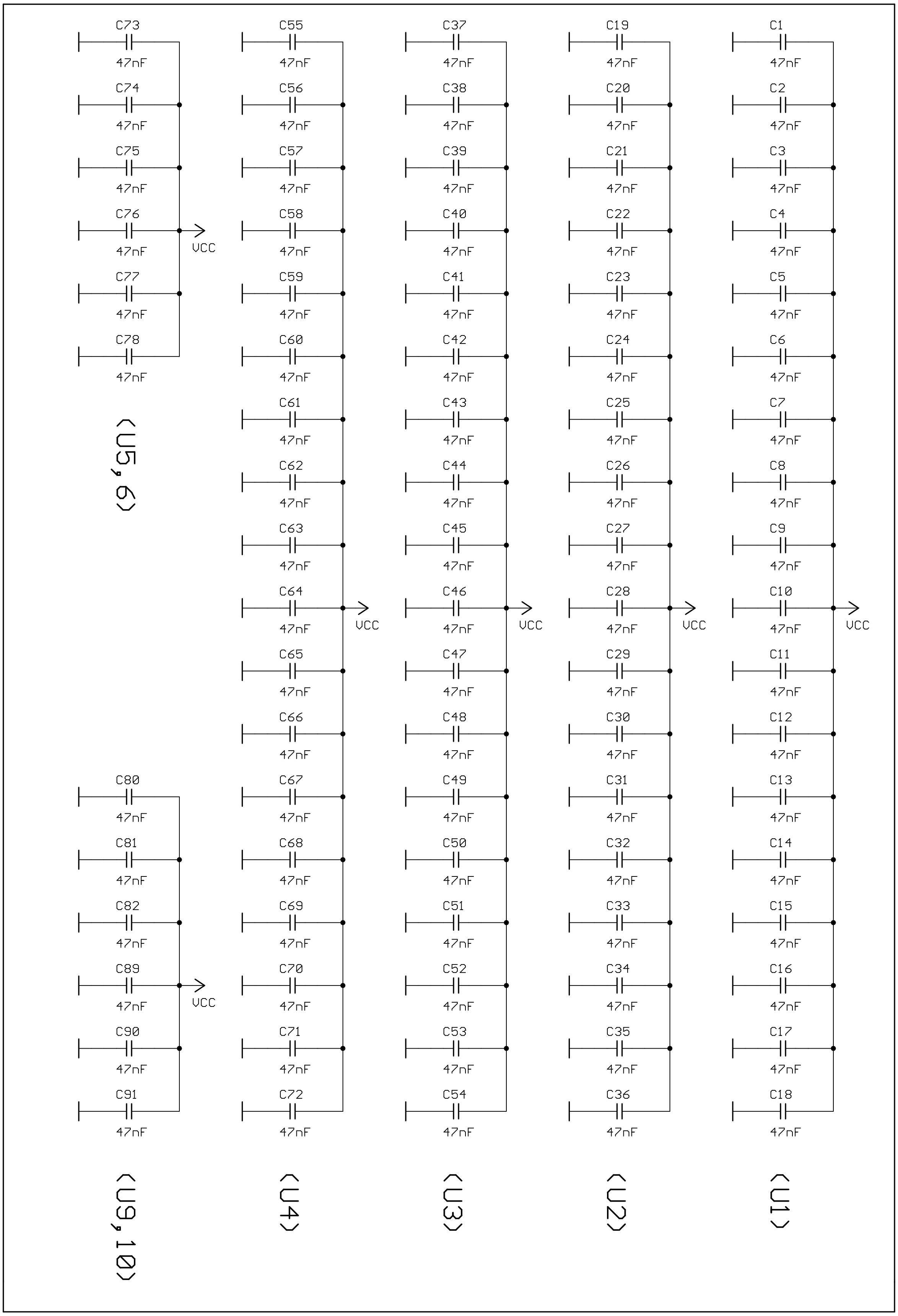


Abbildung B.15: Entkopplungskondensatoren

# Anhang C

# Netzliste

An dieser Stelle wird die vollständige Netzliste des implementierten Prototyping-Boards wiedergegeben. Jeder der durch Leerzeilen getrennten Abschnitte repräsentiert ein Signal, dessen Name zu Beginn des jeweiligen Abschnittes in der ersten Spate „Netz“ erscheint. Jede einzelne Zeile wiederum repräsentiert ein Pin, das mit dem betreffenden Signal beschaltet ist. Die in der Spalte „Blatt“ angegebene Zahl bezeichnet die Nummer des Schaltplan-Blattes, auf dem das Pin zu finden ist. Das erste Blatt ist in Abbildung B.1 dargestellt, die weiteren fortlaufend.

| Netz | Bauteil | Pad | Pin | Blatt |
|---|---|---|---|---|
| +3V_BATT | B1 | +3V | +3V | 3 |
| | C94 | 2 | 2 | 3 |
| | IC1 | 8 | BATT | 3 |
| +3V_PERH | C83 | 2 | 2 | 3 |
| | C84 | 2 | 2 | 3 |
| | C88 | 2 | 2 | 3 |
| | C93 | 2 | 2 | 3 |
| | IC1 | 1 | OUT | 3 |
| | IC4 | 14 | VCC | 3 |
| | U7 | P32 | VCC | 3 |
| | U8 | P32 | VCC | 3 |
| +5V | ST2 | A32 | 32 | 7 |
| | ST2 | B32 | 32 | 7 |
| | ST2 | C32 | 32 | 7 |
| | ST3 | A1 | 1 | 8 |
| | ST3 | B1 | 1 | 8 |
| | ST3 | C1 | 1 | 8 |
| 5V/3V | C79 | 2 | 2 | 1 |
| | C85 | 2 | 2 | 1 |
| | C95 | 1 | 1 | 6 |
| | IC2 | 16 | VCC | 1 |
| | IC3 | 16 | VCC | 1 |
| | IC5 | 3 | IN | 6 |
| | JP14 | 1 | 1 | 1 |
| | JP15 | 1 | 1 | 1 |
| | JP16 | 1 | 1 | 1 |
| | JP17 | 1 | 1 | 1 |
| | JP30 | 1 | 1 | 1 |
| | QG1 | 8 | VCC | 1 |
| | ST1 | A32 | 32 | 6 |
| | ST1 | B32 | 32 | 6 |
| | ST1 | C32 | 32 | 6 |
| | ST4 | A1 | 1 | 9 |
| | ST4 | B1 | 1 | 9 |
| | ST4 | C1 | 1 | 9 |
| | X1 | 1 | KL | 6 |
| A | JP21 | 1 | 1 | 3 |
| | R45 | 2 | 2 | 3 |
| A19 | IC4 | 1 | I0 | 3 |
| | IC4 | 10 | I1 | 3 |
| | IC4 | 9 | I0 | 3 |
| | JP21 | 29 | 29 | 3 |
| | R37 | 1 | 1 | 3 |
| AAA0 | JP21 | 30 | 30 | 3 |
| | U1 | C1 | IO,A13 | 11 |
| | U2 | N1 | IO,A4 | 11 |
| | U5 | P32 | A5 | 4 |
| | U6 | P32 | A5 | 4 |
| | U7 | P28 | A13 | 3 |
| | U8 | P28 | A13 | 3 |
| AAA1 | U1 | D1 | IO | 11 |
| | U3 | A3 | IO | 12 |
| AAA2 | U1 | G1 | IO | 11 |
| | U2 | U2 | IO | 11 |
| | U5 | P24 | IO28 | 4 |
| | U6 | P24 | IO28 | 4 |
| AAA3 | JP21 | 24 | 24 | 3 |
| | U1 | H1 | IO,A10 | 11 |
| | U2 | H1 | IO,A21 | 11 |
| | U5 | P35 | A2 | 4 |
| | U6 | P35 | A2 | 4 |
| | U7 | P23 | A10 | 3 |
| | U8 | P23 | A10 | 3 |
| AAA4 | U1 | J1 | IO | 11 |
| | U2 | R2 | IO | 11 |
| | U5 | P25 | IO29 | 4 |
| | U6 | P25 | IO29 | 4 |
| AAA5 | U1 | H1 | IO,A21 | 11 |
| | U4 | X13 | IO | 10 |
| AAA6 | JP21 | 12 | 12 | 3 |
| | U1 | N1 | IO,A4 | 11 |
| | U2 | H3 | IO | 11 |
| | U5 | P82 | A8 | 4 |
| | U6 | P82 | A8 | 4 |
| | U7 | P8 | A4 | 3 |
| | U8 | P8 | A4 | 3 |
| AAA7 | U1 | P1 | IO | 11 |
| | U2 | G5 | IO,A12 | 11 |
| | U5 | P51 | IO_P1 | 4 |
| | U6 | P51 | IO_P1 | 4 |
| AAA8 | U1 | U1 | IO | 11 |
| | U2 | E2 | IO | 11 |
| | U5 | P58 | IO4 | 4 |
| | U6 | P58 | IO4 | 4 |
| AAA9 | JP21 | 10 | 10 | 3 |
| | U1 | V1 | IO,A3 | 11 |
| | U2 | H1 | IO,A10 | 11 |
| | U5 | P81 | A9 | 4 |
| | U6 | P81 | A9 | 4 |
| | U7 | P9 | A3 | 3 |
| | U8 | P9 | A3 | 3 |
| AAA10 | JP21 | 25 | 25 | 3 |
| | U1 | B2 | IO,A17 | 11 |
| | U2 | H5 | IO | 11 |
| | U5 | P83 | NC/A17 | 4 |
| | U6 | P83 | NC/A17 | 4 |
| | U7 | P30 | A17 | 3 |
| | U8 | P30 | A17 | 3 |
| AAA11 | U1 | C2 | IO | 11 |
| | U3 | A4 | IO | 12 |
| AAA12 | U1 | D2 | IO | 11 |
| | U3 | B3 | IO | 12 |
| AAA13 | U1 | E2 | IO | 11 |
| | U3 | B4 | IO | 12 |
| AAA14 | U1 | F2 | IO | 11 |
| | U3 | B5 | IO | 12 |
| AAA15 | U1 | G2 | IO | 11 |
| | U2 | U3 | IO,A2,CS1 | 11 |
| | U5 | P28 | IO30 | 4 |
| | U6 | P28 | IO30 | 4 |
| AAA16 | U1 | H2 | IO | 11 |
| | U2 | T2 | IO | 11 |
| | U5 | P8 | IO20 | 4 |
| | U6 | P8 | IO20 | 4 |
| AAA17 | U1 | J2 | IO | 11 |
| | U2 | R3 | IO | 11 |
| | U5 | P29 | IO31 | 4 |
| | U6 | P29 | IO31 | 4 |
| AAA18 | JP21 | 20 | 20 | 3 |
| | U1 | K2 | IO,A8 | 11 |
| | U2 | J5 | IO | 11 |
| | U5 | P46 | A12 | 4 |
| | U6 | P46 | A12 | 4 |
| | U7 | P27 | A8 | 3 |
| | U8 | P27 | A8 | 3 |
| AAA19 | JP21 | 18 | 18 | 3 |

| | | | | |
|---|---|---|---|---|
| | U1 | L2 | IO,A7 | 11 |
| | U2 | J1 | IO | 11 |
| | U5 | P48 | A14 | 4 |
| | U6 | P48 | A14 | 4 |
| | U7 | P5 | A7 | 3 |
| | U8 | P5 | A7 | 3 |
| AAA20 | U1 | M2 | IO | 11 |
| | U2 | P2 | IO | 11 |
| | U5 | P9 | IO21 | 4 |
| | U6 | P9 | IO21 | 4 |
| AAA21 | U1 | N2 | IO | 11 |
| | U2 | N3 | IO | 11 |
| | U5 | P31 | LBO# | 4 |
| | U6 | P31 | LBO# | 4 |
| AAA22 | U1 | P2 | IO | 11 |
| | U2 | J4 | IO | 11 |
| | U5 | P85 | ADV/LD# | 4 |
| | U6 | P85 | ADV/LD# | 4 |
| AAA23 | U1 | R2 | IO | 11 |
| | U2 | G1 | IO | 11 |
| | U5 | P63 | IO7 | 4 |
| | U6 | P63 | IO7 | 4 |
| AAA24 | U1 | T2 | IO | 11 |
| | U2 | F2 | IO | 11 |
| | U5 | P79 | IO15 | 4 |
| | U6 | P79 | IO15 | 4 |
| AAA25 | U1 | U2 | IO | 11 |
| | U2 | E3 | IO | 11 |
| | U5 | P56 | IO2 | 4 |
| | U6 | P56 | IO2 | 4 |
| AAA26 | JP21 | 6 | 6 | 3 |
| | U1 | V2 | IO,A1,G7 | 11 |
| | U2 | L5 | IO | 11 |
| | U5 | P36 | A1 | 4 |
| | U6 | P36 | A1 | 4 |
| | U7 | P11 | A1 | 3 |
| | U8 | P11 | A1 | 3 |
| AAA27 | JP21 | 21 | 21 | 3 |
| | U1 | C3 | IO,A15,G8 | 11 |
| | U2 | N2 | IO | 11 |
| | U5 | P100 | A6 | 4 |
| | U6 | P100 | A6 | 4 |
| | U7 | P31 | A15 | 3 |
| | U8 | P31 | A15 | 3 |
| AAA28 | JP21 | 19 | 19 | 3 |
| | U1 | D3 | IO,A14 | 11 |
| | U2 | M4 | IO | 11 |
| | U5 | P33 | A4 | 4 |
| | U6 | P33 | A4 | 4 |
| | U7 | P3 | A14 | 3 |
| | U8 | P3 | A14 | 3 |
| AAA29 | U1 | E3 | IO | 11 |
| | U3 | C5 | IO | 12 |
| AAA30 | U1 | F3 | IO | 11 |
| | U3 | D6 | IO | 12 |
| AAA31 | U1 | G3 | IO | 11 |
| | U2 | V1 | IO,A3 | 11 |
| | U5 | P30 | IO_P4 | 4 |
| | U6 | P30 | IO_P4 | 4 |
| AAA32 | U1 | H3 | IO | 11 |
| | U2 | T3 | IO | 11 |
| | U5 | P12 | IO22 | 4 |
| | U6 | P12 | IO22 | 4 |
| AAA33 | JP21 | 26 | 26 | 3 |
| | U1 | J3 | IO,A11 | 11 |
| | U2 | K5 | IO,A19 | 11 |
| | U5 | P44 | A10 | 4 |
| | U6 | P44 | A10 | 4 |
| | U7 | P25 | A11 | 3 |
| | U8 | P25 | A11 | 3 |
| AAA34 | JP21 | 22 | 22 | 3 |
| | U1 | K3 | IO,A9 | 11 |
| | U2 | K3 | IO,A9 | 11 |
| | U5 | P45 | A11 | 4 |
| | U6 | P45 | A11 | 4 |
| | U7 | P26 | A9 | 3 |
| | U8 | P26 | A9 | 3 |
| AAA35 | JP21 | 16 | 16 | 3 |
| | U1 | L3 | IO,A6 | 11 |
| | U2 | J3 | IO,A11 | 11 |
| | U5 | P47 | A13 | 4 |
| | U6 | P47 | A13 | 4 |
| | U7 | P6 | A6 | 3 |
| | U8 | P6 | A6 | 3 |
| AAA36 | JP21 | 14 | 14 | 3 |
| | U1 | M3 | IO,A5 | 11 |
| | U2 | H4 | IO | 11 |
| | U5 | P49 | A15 | 4 |
| | U6 | P49 | A15 | 4 |
| | U7 | P7 | A5 | 3 |
| | U8 | P7 | A5 | 3 |
| AAA37 | U1 | N3 | IO | 11 |
| | U2 | N4 | IO | 11 |
| | U5 | P1 | IO_P3 | 4 |
| | U6 | P1 | IO_P3 | 4 |
| AAA38 | JP21 | 33 | 33 | 3 |
| | R8 | 2 | 2 | 3 |
| | U1 | P3 | IO | 11 |
| | U2 | K2 | IO,A8 | 11 |
| | U5 | P86 | OE# | 4 |
| | U6 | P86 | OE# | 4 |
| | U7 | P24 | OE# | 3 |
| | U8 | P24 | OE# | 3 |
| AAA39 | U1 | R3 | IO | 11 |
| | U2 | G2 | IO | 11 |
| | U5 | P59 | IO5 | 4 |
| | U6 | P59 | IO5 | 4 |
| AAA40 | U1 | T3 | IO | 11 |
| | U2 | F3 | IO | 11 |
| | U5 | P75 | IO13 | 4 |
| | U6 | P75 | IO13 | 4 |
| AAA41 | JP21 | 8 | 8 | 3 |
| | U1 | U3 | IO,A2,CS1 | 11 |
| | U2 | H2 | IO | 11 |
| | U5 | P50 | A16 | 4 |
| | U6 | P50 | A16 | 4 |
| | U7 | P10 | A2 | 3 |
| | U8 | P10 | A2 | 3 |
| AAA42 | JP21 | 23 | 23 | 3 |
| | U1 | D4 | IO,A16,G1 | 11 |
| | U2 | M5 | IO | 11 |
| | U5 | P99 | A7 | 4 |
| | U6 | P99 | A7 | 4 |
| | U7 | P2 | A16 | 3 |
| | U8 | P2 | A16 | 3 |
| AAA43 | U1 | E4 | IO | 11 |
| | U3 | E6 | IO | 12 |
| AAA44 | R45 | 1 | 1 | 3 |
| | U1 | F4 | IO | 11 |
| | U2 | T14 | IO | 12 |
| AAA45 | R46 | 2 | 2 | 3 |
| | U1 | G4 | IO | 11 |
| | U2 | T15 | IO | 12 |
| AAA46 | U1 | H4 | IO | 11 |
| | U2 | T4 | IO | 11 |
| | U5 | P18 | IO24 | 4 |
| | U6 | P18 | IO24 | 4 |
| AAA47 | U1 | J4 | IO | 11 |
| | U2 | R4 | IO | 11 |
| | U5 | P2 | IO16 | 4 |
| | U6 | P2 | IO16 | 4 |
| AAA48 | JP21 | 27 | 27 | 3 |
| | U1 | K4 | IO,A18 | 11 |
| | U2 | J2 | IO | 11 |
| | U5 | P84 | NC/A18 | 4 |
| | U6 | P84 | NC/A18 | 4 |
| | U7 | P1 | A18 | 3 |
| | U8 | P1 | A18 | 3 |
| AAA49 | U1 | L4 | IO,A20 | 11 |
| | U4 | D20 | IO | 14 |
| AAA50 | U1 | M4 | IO | 11 |
| | U2 | P3 | IO | 11 |
| | U5 | P13 | IO23 | 4 |
| | U6 | P13 | IO23 | 4 |
| AAA51 | U1 | N4 | IO | 11 |
| | U2 | N5 | IO | 11 |
| | U5 | P3 | IO17 | 4 |
| | U6 | P3 | IO17 | 4 |
| AAA52 | U1 | P4 | IO | 11 |
| | U2 | K4 | IO,A18 | 11 |
| | U5 | P87 | CEN# | 4 |
| | U6 | P87 | CEN# | 4 |
| AAA53 | U1 | R4 | IO | 11 |
| | U2 | G3 | IO | 11 |
| | U5 | P57 | IO3 | 4 |
| | U6 | P57 | IO3 | 4 |
| AAA54 | U1 | T4 | IO | 11 |
| | U2 | F4 | IO | 11 |
| | U5 | P73 | IO11 | 4 |
| | U6 | P73 | IO11 | 4 |
| AAA55 | U1 | F5 | IO | 11 |
| | U2 | U14 | IO | 12 |
| AAA56 | JP21 | 28 | 28 | 3 |
| | U1 | G5 | IO,A12 | 11 |
| | U2 | M3 | IO,A5 | 11 |
| | U5 | P34 | A3 | 4 |
| | U6 | P34 | A3 | 4 |
| | U7 | P4 | A12 | 3 |
| | U8 | P4 | A12 | 3 |
| AAA57 | U1 | H5 | IO | 11 |
| | U2 | U1 | IO | 11 |
| | U5 | P22 | IO26 | 4 |
| | U6 | P22 | IO26 | 4 |
| AAA58 | U1 | J5 | IO | 11 |
| | U2 | R5 | IO | 11 |
| | U5 | P6 | IO18 | 4 |
| | U6 | P6 | IO18 | 4 |
| AAA59 | R37 | 2 | 2 | 3 |
| | U1 | K5 | IO,A19 | 11 |
| | U2 | M2 | IO | 11 |
| | U5 | P97 | CE2 | 4 |
| | U6 | P98 | CE1# | 4 |
| AAA60 | U1 | L5 | IO | 11 |
| | U2 | P5 | IO | 11 |
| | U5 | P23 | IO27 | 4 |
| | U6 | P23 | IO27 | 4 |
| AAA61 | U1 | M5 | IO | 11 |
| | U2 | P4 | IO | 11 |
| | U5 | P19 | IO25 | 4 |
| | U6 | P19 | IO25 | 4 |
| AAA62 | U1 | N5 | IO | 11 |
| | U2 | P1 | IO | 11 |
| | U5 | P7 | IO19 | 4 |
| | U6 | P7 | IO19 | 4 |
| AAA63 | R9 | 1 | 1 | 4 |
| | U1 | P5 | IO | 11 |
| | U2 | L3 | IO,A6 | 11 |
| | U5 | P92 | CE2# | 4 |
| | U6 | P92 | CE2# | 4 |
| AAA64 | U1 | R5 | IO | 11 |
| | U2 | G4 | IO | 11 |
| | U5 | P53 | IO1 | 4 |
| | U6 | P53 | IO1 | 4 |
| B | JP21 | 2 | 2 | 3 |
| | R46 | 1 | 1 | 3 |
| BBB1 | ST1 | B30 | 30 | 6 |
| | U1 | X3 | IO | 6 |
| BBB2 | ST1 | C29 | 29 | 6 |
| | U1 | X4 | IO | 6 |
| BBB3 | ST1 | B26 | 26 | 6 |
| | U1 | X7 | IO | 6 |
| BBB4 | ST1 | A25 | 25 | 6 |
| | U1 | X8 | IO | 6 |
| BBB5 | ST1 | A23 | 23 | 6 |
| | U1 | X9 | IO | 6 |
| BBB6 | ST1 | B19 | 19 | 6 |
| | U1 | X12 | IO | 6 |
| BBB7 | ST1 | B17 | 17 | 6 |
| | U1 | X13 | IO | 6 |
| BBB8 | ST1 | B15 | 15 | 6 |
| | U1 | X14 | IO,CS0 | 6 |
| BBB9 | ST1 | C11 | 11 | 6 |
| | U1 | X17 | IO | 6 |

| | | | | |
|---|---|---|---|---|
| BBB10 | ST1 | B10 | 10 | 6 |
| | U1 | X18 | IO | 6 |
| BBB11 | JP21 | 4 | 4 | 3 |
| | U1 | W2 | IO,A0,WS | 6 |
| | U2 | L4 | IO,A20 | 11 |
| | U5 | P37 | A0 | 4 |
| | U6 | P37 | A0 | 4 |
| | U7 | P12 | A0 | 3 |
| | U8 | P12 | A0 | 3 |
| BBB12 | ST1 | C30 | 30 | 6 |
| | U1 | W3 | IO | 6 |
| BBB13 | ST1 | A30 | 30 | 6 |
| | U1 | W4 | IO | 6 |
| BBB14 | ST1 | A29 | 29 | 6 |
| | U1 | W5 | IO | 6 |
| BBB15 | ST1 | A28 | 28 | 6 |
| | U1 | W6 | IO | 6 |
| BBB16 | ST1 | C26 | 26 | 6 |
| | U1 | W7 | IO | 6 |
| BBB17 | JP21 | 7 | 7 | 3 |
| | U1 | W8 | IO,D2 | 6 |
| | U2 | B2 | IO,A17 | 11 |
| | U5 | P80 | IO_P2 | 4 |
| | U6 | P80 | IO_P2 | 4 |
| | U7 | P15 | IO2 | 3 |
| | U8 | P15 | IO2 | 3 |
| BBB18 | ST1 | C23 | 23 | 6 |
| | U1 | W9 | IO | 6 |
| BBB19 | JP21 | 9 | 9 | 3 |
| | U1 | W10 | IO,D3 | 6 |
| | U2 | C1 | IO,A13 | 11 |
| | U5 | P78 | IO14 | 4 |
| | U6 | P78 | IO14 | 4 |
| | U7 | P17 | IO3 | 3 |
| | U8 | P17 | IO3 | 3 |
| BBB20 | ST1 | A21 | 21 | 6 |
| | U1 | W11 | IO | 6 |
| BBB21 | ST1 | C19 | 19 | 6 |
| | U1 | W12 | IO | 6 |
| BBB22 | ST1 | C17 | 17 | 6 |
| | U1 | W13 | IO | 6 |
| BBB23 | ST1 | C15 | 15 | 6 |
| | U1 | W14 | IO | 6 |
| BBB24 | ST1 | C13 | 13 | 6 |
| | U1 | W15 | IO | 6 |
| BBB25 | ST1 | C12 | 12 | 6 |
| | U1 | W16 | IO | 6 |
| BBB26 | JP21 | 15 | 15 | 3 |
| | U1 | W17 | IO,D6 | 6 |
| | U2 | C2 | IO | 11 |
| | U5 | P74 | IO12 | 4 |
| | U6 | P74 | IO12 | 4 |
| | U7 | P20 | IO6 | 3 |
| | U8 | P20 | IO6 | 3 |
| BBB29 | U1 | V5 | IO,RCLK,RB | 6 |
| | U4 | X14 | IO,CS0 | 10 |
| BBB30 | ST1 | B28 | 28 | 6 |
| | U1 | V6 | IO | 6 |
| BBB31 | ST1 | A27 | 27 | 6 |
| | U1 | V7 | IO | 6 |
| BBB32 | ST1 | B25 | 25 | 6 |
| | U1 | V8 | IO | 6 |
| BBB33 | ST1 | A24 | 24 | 6 |
| | U1 | V9 | IO | 6 |
| BBB34 | ST1 | A22 | 22 | 6 |
| | U1 | V10 | IO,RS | 6 |
| BBB35 | JP21 | 11 | 11 | 3 |
| | U1 | V11 | IO,D4 | 6 |
| | U2 | D3 | IO,A14 | 11 |
| | U5 | P62 | IO6 | 4 |
| | U6 | P62 | IO6 | 4 |
| | U7 | P18 | IO4 | 3 |
| | U8 | P18 | IO4 | 3 |
| BBB36 | ST1 | A20 | 20 | 6 |
| | U1 | V12 | IO | 6 |
| BBB37 | ST1 | A18 | 18 | 6 |
| | U1 | V13 | IO | 6 |
| BBB38 | ST1 | A16 | 16 | 6 |
| | U1 | V14 | IO | 6 |
| BBB39 | ST1 | A14 | 14 | 6 |
| | U1 | V15 | IO | 6 |
| BBB40 | ST1 | A13 | 13 | 6 |
| | U1 | V16 | IO | 6 |
| BBB41 | ST1 | A12 | 12 | 6 |
| | U1 | V17 | IO | 6 |
| BBB43 | ST1 | B29 | 29 | 6 |
| | U1 | U5 | IO | 6 |
| BBB44 | JP21 | 5 | 5 | 3 |
| | U1 | U6 | IO,D1 | 6 |
| | U2 | E4 | IO | 11 |
| | U5 | P52 | IO0 | 4 |
| | U6 | P52 | IO0 | 4 |
| | U7 | P14 | IO1 | 3 |
| | U8 | P14 | IO1 | 3 |
| BBB45 | ST1 | B27 | 27 | 6 |
| | U1 | U7 | IO | 6 |
| BBB46 | ST1 | C25 | 25 | 6 |
| | U1 | U8 | IO | 6 |
| BBB47 | ST1 | B24 | 24 | 6 |
| | U1 | U9 | IO | 6 |
| BBB48 | ST1 | B22 | 22 | 6 |
| | U1 | U10 | IO | 6 |
| BBB49 | ST1 | B21 | 21 | 6 |
| | U1 | U11 | IO | 6 |
| BBB50 | ST1 | C20 | 20 | 6 |
| | U1 | U12 | IO | 6 |
| BBB51 | ST1 | B18 | 18 | 6 |
| | U1 | U13 | IO | 6 |
| BBB52 | ST1 | B16 | 16 | 6 |
| | U1 | U14 | IO | 6 |
| BBB53 | ST1 | B14 | 14 | 6 |
| | U1 | U15 | IO | 6 |
| BBB54 | ST1 | B13 | 13 | 6 |
| | U1 | U16 | IO | 6 |
| BBB55 | ST1 | C28 | 28 | 6 |
| | U1 | T6 | IO | 6 |
| BBB56 | ST1 | C27 | 27 | 6 |
| | U1 | T7 | IO | 6 |
| BBB57 | ST1 | A26 | 26 | 6 |
| | U1 | T8 | IO | 6 |
| BBB58 | ST1 | C24 | 24 | 6 |
| | U1 | T9 | IO | 6 |
| BBB59 | ST1 | C22 | 22 | 6 |
| | U1 | T10 | IO | 6 |
| BBB60 | ST1 | C21 | 21 | 6 |
| | U1 | T11 | IO | 6 |
| BBB61 | JP21 | 13 | 13 | 3 |
| | U1 | T12 | IO,D5 | 6 |
| | U2 | F5 | IO | 11 |
| | U5 | P69 | IO9 | 4 |
| | U6 | P69 | IO9 | 4 |
| | U7 | P19 | IO5 | 3 |
| | U8 | P19 | IO5 | 3 |
| BBB62 | ST1 | C18 | 18 | 6 |
| | U1 | T13 | IO | 6 |
| BBB63 | ST1 | C16 | 16 | 6 |
| | U1 | T14 | IO | 6 |
| BBB64 | ST1 | C14 | 14 | 6 |
| | U1 | T15 | IO | 6 |
| CCC1 | U1 | A4 | IO | 10 |
| | U2 | X8 | IO | 12 |
| | U3 | A8 | IO | 12 |
| | U4 | X4 | IO | 10 |
| CCC2 | U1 | A7 | IO | 10 |
| | U2 | X9 | IO | 12 |
| | U3 | A9 | IO | 12 |
| | U4 | X7 | IO | 10 |
| CCC3 | U1 | A8 | IO | 10 |
| | U2 | X12 | IO | 12 |
| | U3 | A12 | IO | 12 |
| | U4 | X8 | IO | 10 |
| CCC4 | U1 | A9 | IO | 10 |
| | U2 | X13 | IO | 12 |
| | U3 | A13 | IO | 12 |
| | U4 | X9 | IO | 10 |
| CCC6 | U1 | A13 | IO | 10 |
| | U4 | C19 | IO,HDC | 14 |
| CCC7 | U1 | A14 | IO | 10 |
| | U4 | C20 | IO,LDC | 14 |
| CCC8 | U1 | A17 | IO | 10 |
| | U4 | X17 | IO | 10 |
| CCC9 | U1 | A18 | IO | 10 |
| | U4 | X18 | IO | 10 |
| CCC10 | U1 | B3 | IO | 10 |
| | U2 | W6 | IO | 12 |
| | U3 | B6 | IO | 12 |
| | U4 | W3 | IO | 10 |
| CCC11 | U1 | B4 | IO | 10 |
| | U2 | W7 | IO | 12 |
| | U3 | B7 | IO | 12 |
| | U4 | W4 | IO | 10 |
| CCC12 | U1 | B5 | IO | 10 |
| | U2 | W8 | IO,D2 | 12 |
| | U3 | B8 | IO | 12 |
| | U4 | W5 | IO | 10 |
| CCC13 | U1 | B6 | IO | 10 |
| | U2 | W9 | IO | 12 |
| | U3 | B9 | IO | 12 |
| | U4 | W6 | IO | 10 |
| CCC14 | U1 | B7 | IO | 10 |
| | U2 | W10 | IO,D3 | 12 |
| | U3 | B10 | IO | 12 |
| | U4 | W7 | IO | 10 |
| CCC15 | U1 | B8 | IO | 10 |
| | U2 | W11 | IO | 12 |
| | U3 | B11 | IO | 12 |
| | U4 | W8 | IO,D2 | 10 |
| CCC16 | U1 | B9 | IO | 10 |
| | U2 | W12 | IO | 12 |
| | U3 | B12 | IO | 12 |
| | U4 | W9 | IO | 10 |
| CCC17 | U1 | B10 | IO | 10 |
| | U2 | W13 | IO | 12 |
| | U3 | B13 | IO | 12 |
| | U4 | W10 | IO,D3 | 10 |
| CCC18 | U1 | B11 | IO | 10 |
| | U2 | W14 | IO | 12 |
| | U3 | B14 | IO | 12 |
| | U4 | W11 | IO | 10 |
| CCC19 | U1 | B12 | IO | 10 |
| | U2 | V6 | IO | 12 |
| | U3 | C6 | IO | 12 |
| | U4 | W12 | IO | 10 |
| CCC20 | U1 | B13 | IO | 10 |
| | U4 | W13 | IO | 10 |
| CCC21 | U1 | B14 | IO | 10 |
| | U4 | W14 | IO | 10 |
| CCC22 | U1 | B15 | IO | 10 |
| | U4 | W15 | IO | 10 |
| CCC23 | U1 | B16 | IO | 10 |
| | U4 | W16 | IO | 10 |
| CCC24 | U1 | B17 | IO | 10 |
| | U4 | W17 | IO,D6 | 10 |

| | | | | |
|---|---|---|---|---|
| CCC25 | U1 | B18 | IO | 10 |
| | U4 | V17 | IO | 10 |
| CCC28 | U1 | C5 | IO | 10 |
| | U2 | V8 | IO | 12 |
| | U3 | C8 | IO | 12 |
| | U4 | V5 | IO,RCLK,RB | 10 |
| CCC29 | U1 | C6 | IO | 10 |
| | U2 | V9 | IO | 12 |
| | U3 | C9 | IO | 12 |
| | U4 | V6 | IO | 10 |
| CCC31 | U1 | C8 | IO | 10 |
| | U2 | V10 | IO,RS | 12 |
| | U3 | C10 | IO | 12 |
| | U4 | V8 | IO | 10 |
| CCC32 | U1 | C9 | IO | 10 |
| | U2 | V11 | IO,D4 | 12 |
| | U3 | C11 | IO | 12 |
| | U4 | V9 | IO | 10 |
| CCC33 | U1 | C10 | IO | 10 |
| | U2 | V12 | IO | 12 |
| | U3 | C12 | IO | 12 |
| | U4 | V10 | IO,RS | 10 |
| CCC34 | U1 | C11 | IO | 10 |
| | U2 | V13 | IO | 12 |
| | U3 | C13 | IO | 12 |
| | U4 | V11 | IO,D4 | 10 |
| CCC35 | U1 | C12 | IO | 10 |
| | U2 | V14 | IO | 12 |
| | U3 | C14 | IO | 12 |
| | U4 | V12 | IO | 10 |
| CCC36 | U1 | C13 | IO | 10 |
| | U4 | V13 | IO | 10 |
| CCC37 | U1 | C14 | IO | 10 |
| | U4 | V14 | IO | 10 |
| CCC38 | U1 | C15 | IO | 10 |
| | U4 | V15 | IO | 10 |
| CCC39 | U1 | C16 | IO | 10 |
| | U4 | V16 | IO | 10 |
| CCC42 | U1 | D6 | IO | 10 |
| | U2 | U7 | IO | 12 |
| | U3 | D7 | IO | 12 |
| | U4 | U6 | IO,D1 | 10 |
| CCC43 | U1 | D7 | IO | 10 |
| | U2 | U8 | IO | 12 |
| | U3 | D8 | IO | 12 |
| | U4 | U7 | IO | 10 |
| CCC44 | U1 | D8 | IO | 10 |
| | U2 | U9 | IO | 12 |
| | U3 | D9 | IO | 12 |
| | U4 | U8 | IO | 10 |
| CCC45 | U1 | D9 | IO | 10 |
| | U2 | U10 | IO | 12 |
| | U3 | D10 | IO | 12 |
| | U4 | U9 | IO | 10 |
| CCC46 | U1 | D10 | IO | 10 |
| | U2 | U11 | IO | 12 |
| | U3 | D11 | IO | 12 |
| | U4 | U10 | IO | 10 |
| CCC47 | U1 | D11 | IO | 10 |
| | U2 | U12 | IO | 12 |
| | U3 | D12 | IO | 12 |
| | U4 | U11 | IO | 10 |
| CCC48 | U1 | D12 | IO | 10 |
| | U2 | U13 | IO | 12 |
| | U3 | D13 | IO | 12 |
| | U4 | U12 | IO | 10 |
| CCC49 | U1 | D13 | IO | 10 |
| | U4 | U13 | IO | 10 |
| CCC50 | U1 | D14 | IO | 10 |
| | U4 | U14 | IO | 10 |
| CCC51 | U1 | D15 | IO | 10 |
| | U4 | U15 | IO | 10 |
| CCC52 | U1 | D16 | IO | 10 |
| | U4 | U16 | IO | 10 |
| CCC53 | U1 | E6 | IO | 10 |
| | U2 | T7 | IO | 12 |
| | U3 | E7 | IO | 12 |
| | U4 | T6 | IO | 10 |
| CCC54 | U1 | E7 | IO | 10 |
| | U2 | T8 | IO | 12 |
| | U3 | E8 | IO | 12 |
| | U4 | T7 | IO | 10 |
| CCC55 | U1 | E8 | IO | 10 |
| | U2 | T9 | IO | 12 |
| | U3 | E9 | IO | 12 |
| | U4 | T8 | IO | 10 |
| CCC56 | U1 | E9 | IO | 10 |
| | U2 | T10 | IO | 12 |
| | U3 | E10 | IO | 12 |
| | U4 | T9 | IO | 10 |
| CCC57 | U1 | E10 | IO | 10 |
| | U2 | T11 | IO | 12 |
| | U3 | E11 | IO | 12 |
| | U4 | T10 | IO | 10 |
| CCC58 | U1 | E11 | IO | 10 |
| | U2 | T12 | IO,D5 | 12 |
| | U3 | E12 | IO | 12 |
| | U4 | T11 | IO | 10 |
| CCC59 | U1 | E12 | IO | 10 |
| | U2 | T13 | IO | 12 |
| | U3 | E13 | IO | 12 |
| | U4 | T12 | IO,D5 | 10 |
| CCC60 | U1 | E13 | IO | 10 |
| | U4 | T13 | IO | 10 |
| CCC61 | U1 | E14 | IO | 10 |
| | U4 | T14 | IO | 10 |
| CCC62 | U1 | E15 | IO | 10 |
| | U4 | T15 | IO | 10 |
| CCLK1 | J1 | 7 | CCLK | 2 |
| | JP1 | 1 | 1 | 2 |
| | JP39 | 2 | 2 | 2 |
| | R44 | 1 | 1 | 2 |
| | U1 | V3 | CCLK | 2 |
| CCLK2 | J2 | 7 | CCLK | 2 |
| | JP1 | 2 | 2 | 2 |
| | JP2 | 1 | 1 | 2 |
| | U2 | V3 | CCLK | 2 |
| CCLK3 | J3 | 7 | CCLK | 2 |
| | JP2 | 2 | 2 | 2 |
| | JP6 | 1 | 1 | 2 |
| | U3 | V3 | CCLK | 2 |
| CCLK4 | J4 | 7 | CCLK | 2 |
| | JP6 | 2 | 2 | 2 |
| | U4 | V3 | CCLK | 2 |
| CLK1 | R15 | 1 | 1 | 1 |
| | U2 | B19 | IO,G3 | 7 |
| | U2 | C17 | IO,G2 | 7 |
| CLK2 | R16 | 1 | 1 | 1 |
| | U2 | W18 | IO,G5 | 12 |
| | U2 | X20 | IO,G4 | 7 |
| CLK3 | R17 | 1 | 1 | 1 |
| | U3 | B19 | IO,G3 | 12 |
| | U3 | C17 | IO,G2 | 12 |
| CLK4 | R18 | 1 | 1 | 1 |
| | U3 | W18 | IO,G5 | 8 |
| | U3 | X20 | IO,G4 | 8 |
| CLK5 | R19 | 1 | 1 | 1 |
| | U2 | C3 | IO,A15,G8 | 11 |
| | U2 | D4 | IO,A16,G1 | 11 |
| CLK6 | R20 | 1 | 1 | 1 |
| | U5 | P89 | CLK | 4 |
| | U6 | P89 | CLK | 4 |
| CLK7 | R21 | 1 | 1 | 1 |
| | U1 | W18 | IO,G5 | 6 |
| | U1 | X20 | IO,G4 | 6 |
| CLK8 | R22 | 1 | 1 | 1 |
| | U1 | B19 | IO,G3 | 10 |
| | U1 | C17 | IO,G2 | 10 |
| CLK9 | R23 | 1 | 1 | 1 |
| | U4 | W18 | IO,G5 | 10 |
| | U4 | X20 | IO,G4 | 14 |
| CLK10 | R24 | 1 | 1 | 1 |
| | U4 | B19 | IO,G3 | 9 |
| | U4 | C17 | IO,G2 | 9 |
| CLK11 | R25 | 1 | 1 | 1 |
| | U3 | C3 | IO,A15,G8 | 12 |
| | U3 | D4 | IO,A16,G1 | 12 |
| CLK12 | R26 | 1 | 1 | 1 |
| | U4 | C3 | IO,A15,G8 | 12 |
| | U4 | D4 | IO,A16,G1 | 12 |
| CLK13 | R27 | 1 | 1 | 1 |
| | ST3 | A23 | 23 | 8 |
| | U4 | V2 | IO,A1,G7 | 12 |
| CLK14 | R28 | 1 | 1 | 1 |
| | U10 | 17 | CLK_R | 5 |
| | U10 | 22 | CLK_L | 5 |
| CLK15 | R29 | 1 | 1 | 1 |
| | U9 | 17 | CLK_R | 5 |
| | U9 | 22 | CLK_L | 5 |
| CLK16 | R30 | 1 | 1 | 1 |
| | U2 | V2 | IO,A1,G7 | 11 |
| | U3 | V2 | IO,A1,G7 | 12 |
| CLK17 | R31 | 1 | 1 | 1 |
| | ST4 | A23 | 23 | 9 |
| CLKI1 | J1 | 16 | CLKI | 2 |
| | R11 | 1 | 1 | 1 |
| CLKI2 | J2 | 16 | CLKI | 2 |
| | R12 | 1 | 1 | 1 |
| CLKI3 | J3 | 16 | CLKI | 2 |
| | R13 | 1 | 1 | 1 |
| CLKI4 | J4 | 16 | CLKI | 2 |
| | R14 | 1 | 1 | 1 |
| CLKO1 | IC3 | 15 | D4 | 1 |
| | J1 | 18 | CLKO | 2 |
| CLKO2 | IC3 | 14 | D5 | 1 |
| | J2 | 18 | CLKO | 2 |
| CLKO3 | IC3 | 13 | D6 | 1 |
| | J3 | 18 | CLKO | 2 |
| CLKO4 | IC3 | 12 | D7 | 1 |
| | J4 | 18 | CLKO | 2 |
| CLK_BNC | IC2 | 3 | D1 | 1 |
| | J5 | 1 | CENTER | 1 |
| CLK_EXT1 | IC2 | 2 | D2 | 1 |
| | ST1 | A10 | 10 | 6 |
| CLK_EXT2 | IC2 | 1 | D3 | 1 |
| | ST2 | A10 | 10 | 7 |
| CLK_QG | IC2 | 4 | D0 | 1 |
| | QG1 | 5 | OUT | 1 |
| CLK_SRC | IC2 | 5 | Y | 1 |
| | IC3 | 4 | D0 | 1 |
| | IC6 | 10 | IN_A | 1 |
| CS# | IC1 | 5 | CE#_IN | 3 |
| | JP21 | 34 | 34 | 3 |
| | R38 | 1 | 1 | 3 |
| DDD0 | R38 | 2 | 2 | 3 |
| | U1 | C20 | IO,LDC | 6 |
| | U2 | B10 | IO | 7 |
| DDD1 | U1 | D20 | IO | 6 |
| | U4 | P20 | IO | 14 |
| DDD2 | U1 | G20 | IO | 6 |
| | U4 | N20 | IO | 14 |
| DDD3 | J1 | 6 | TRIG | 2 |
| | J4 | 6 | TRIG | 2 |
| | U1 | H20 | IO | 6 |
| | U4 | H20 | IO | 14 |

| | | | | |
|---|---|---|---|---|
| DDD5 | ST1 | C1 | 1 | 6 |
| | U1 | M20 | IO | 6 |
| DDD6 | ST1 | B3 | 3 | 6 |
| | U1 | N20 | IO | 6 |
| DDD7 | ST1 | A5 | 5 | 6 |
| | U1 | P20 | IO | 6 |
| DDD8 | ST1 | B8 | 8 | 6 |
| | U1 | U20 | IO | 6 |
| DDD9 | ST1 | A8 | 8 | 6 |
| | U1 | V20 | IO | 6 |
| DDD11 | R36 | 2 | 2 | 3 |
| | U1 | C19 | IO,HDC | 6 |
| | U2 | L2 | IO,A7 | 11 |
| | U5 | P88 | R/W# | 4 |
| | U6 | P88 | R/W# | 4 |
| DDD12 | U1 | D19 | IO | 6 |
| | U4 | T19 | IO | 14 |
| DDD13 | U1 | E19 | IO | 6 |
| | U4 | R19 | IO | 14 |
| DDD14 | U1 | F19 | IO | 6 |
| | U4 | P19 | IO | 14 |
| DDD15 | U1 | G19 | IO | 6 |
| | U4 | N19 | IO | 14 |
| DDD16 | U1 | H19 | IO | 6 |
| | U4 | M19 | IO | 14 |
| DDD17 | U1 | J19 | IO | 6 |
| | U4 | L19 | IO | 14 |
| DDD19 | JP28 | 1 | 1 | 6 |
| | U1 | L19 | IO | 6 |
| DDD20 | ST1 | A2 | 2 | 6 |
| | U1 | M19 | IO | 6 |
| DDD21 | ST1 | C3 | 3 | 6 |
| | U1 | N19 | IO | 6 |
| DDD22 | ST1 | B5 | 5 | 6 |
| | U1 | P19 | IO | 6 |
| DDD23 | ST1 | C6 | 6 | 6 |
| | U1 | R19 | IO | 6 |
| DDD25 | ST1 | B9 | 9 | 6 |
| | U1 | U19 | IO | 6 |
| DDD26 | ST1 | C8 | 8 | 6 |
| | U1 | V19 | IO | 6 |
| DDD27 | JP21 | 17 | 17 | 3 |
| | U1 | W19 | IO,D7 | 6 |
| | U2 | D1 | IO | 11 |
| | U5 | P72 | IO10 | 4 |
| | U6 | P72 | IO10 | 4 |
| | U7 | P21 | IO7 | 3 |
| | U8 | P21 | IO7 | 3 |
| DDD28 | U1 | D18 | IO | 6 |
| | U4 | T18 | IO | 14 |
| DDD29 | U1 | E18 | IO | 6 |
| | U4 | R18 | IO | 14 |
| DDD30 | U1 | F18 | IO | 6 |
| | U4 | P18 | IO | 14 |
| DDD31 | U1 | G18 | IO | 6 |
| | U4 | N18 | IO | 14 |
| DDD32 | U1 | H18 | IO | 6 |
| | U4 | M18 | IO | 14 |
| DDD33 | U1 | J18 | IO | 6 |
| | U4 | L18 | IO | 14 |
| DDD34 | JP9 | 1 | 1 | 6 |
| | U1 | K18 | IO | 6 |
| DDD35 | JP27 | 1 | 1 | 6 |
| | U1 | L18 | IO | 6 |
| DDD36 | JP22 | 1 | 1 | 6 |
| | U1 | M18 | IO | 6 |
| DDD37 | ST1 | A4 | 4 | 6 |
| | U1 | N18 | IO | 6 |
| DDD38 | ST1 | C5 | 5 | 6 |
| | U1 | P18 | IO | 6 |
| DDD39 | ST1 | A7 | 7 | 6 |
| | U1 | R18 | IO | 6 |
| DDD40 | ST1 | B11 | 11 | 6 |
| | U1 | T18 | IO | 6 |
| DDD41 | ST1 | C10 | 10 | 6 |
| | U1 | U18 | IO | 6 |
| DDD42 | U1 | E17 | IO | 6 |
| | U4 | R17 | IO | 14 |
| DDD43 | U1 | F17 | IO | 6 |
| | U4 | P17 | IO | 14 |
| DDD44 | U1 | G17 | IO | 6 |
| | U4 | N17 | IO | 14 |
| DDD45 | U1 | H17 | IO | 6 |
| | U4 | M17 | IO | 14 |
| DDD46 | U1 | J17 | IO | 6 |
| | U4 | L17 | IO | 14 |
| DDD47 | JP18 | 1 | 1 | 6 |
| | U1 | K17 | IO | 6 |
| DDD48 | ST1 | A1 | 1 | 6 |
| | U1 | L17 | IO | 6 |
| DDD49 | ST1 | C2 | 2 | 6 |
| | U1 | M17 | IO | 6 |
| DDD50 | ST1 | B4 | 4 | 6 |
| | U1 | N17 | IO | 6 |
| DDD51 | ST1 | A6 | 6 | 6 |
| | U1 | P17 | IO | 6 |
| DDD52 | ST1 | B7 | 7 | 6 |
| | U1 | R17 | IO | 6 |
| DDD53 | ST1 | B12 | 12 | 6 |
| | U1 | T17 | IO | 6 |
| DDD54 | U1 | F16 | IO | 6 |
| | U4 | P16 | IO | 14 |
| DDD55 | U1 | G16 | IO | 6 |
| | U4 | N16 | IO | 14 |
| DDD56 | U1 | H16 | IO | 6 |
| | U4 | M16 | IO | 14 |
| DDD57 | U1 | J16 | IO | 6 |
| | U4 | L16 | IO | 14 |
| DDD58 | JP20 | 1 | 1 | 6 |
| | U1 | K16 | IO | 6 |
| DDD59 | ST1 | B1 | 1 | 6 |
| | U1 | L16 | IO | 6 |
| DDD60 | ST1 | A3 | 3 | 6 |
| | U1 | M16 | IO | 6 |
| DDD61 | ST1 | C4 | 4 | 6 |
| | U1 | N16 | IO | 6 |
| DDD62 | ST1 | B6 | 6 | 6 |
| | U1 | P16 | IO | 6 |
| DDD63 | ST1 | C7 | 7 | 6 |
| | U1 | R16 | IO | 6 |
| DIN1 | J1 | 11 | DIN | 2 |
| | JP21 | 3 | 3 | 3 |
| | U1 | V4 | IO,DO,DIN | 6 |
| | U2 | D2 | IO | 11 |
| | U5 | P68 | IO8 | 4 |
| | U6 | P68 | IO8 | 4 |
| | U7 | P13 | IO0 | 3 |
| | U8 | P13 | IO0 | 3 |
| DIN2 | J2 | 11 | DIN | 2 |
| | U1 | X1 | IO,DOUT,G6 | 6 |
| | U2 | V4 | IO,DO,DIN | 12 |
| DIN3 | J3 | 11 | DIN | 2 |
| | U2 | X1 | IO,DOUT,G6 | 12 |
| | U3 | V4 | IO,DO,DIN | 8 |
| DIN4 | J4 | 11 | DIN | 2 |
| | U3 | X1 | IO,DOUT,G6 | 8 |
| | U4 | V4 | IO,DO,DIN | 10 |
| DONE | J1 | 9 | DONE | 2 |
| | JP10 | 1 | 1 | 2 |
| | R2 | 1 | 1 | 2 |
| | U1 | V18 | DONE | 2 |
| DOUT4 | JP38 | 1 | 1 | 10 |
| | U4 | X1 | IO,DOUT,G6 | 10 |
| EEE8 | U2 | X17 | IO | 12 |
| | U3 | A17 | IO | 12 |
| EEE9 | U2 | X18 | IO | 12 |
| | U3 | A18 | IO | 12 |
| EEE22 | U2 | W15 | IO | 12 |
| | U3 | B15 | IO | 12 |
| EEE23 | U2 | W16 | IO | 12 |
| | U3 | B16 | IO | 12 |
| EEE24 | U2 | W17 | IO,D6 | 12 |
| | U3 | B17 | IO | 12 |
| EEE25 | U2 | V17 | IO | 12 |
| | U3 | B18 | IO | 12 |
| EEE38 | U2 | V15 | IO | 12 |
| | U3 | C15 | IO | 12 |
| EEE39 | U2 | V16 | IO | 12 |
| | U3 | C16 | IO | 12 |
| EEE50 | U3 | D14 | IO | 12 |
| | U4 | V20 | IO | 14 |
| EEE51 | U2 | U15 | IO | 12 |
| | U3 | D15 | IO | 12 |
| EEE52 | U2 | U16 | IO | 12 |
| | U3 | D16 | IO | 12 |
| EEE61 | U3 | E14 | IO | 12 |
| | U4 | W19 | IO,D7 | 14 |
| EEE62 | U3 | E15 | IO | 12 |
| | U4 | V19 | IO | 14 |
| GGG0 | U2 | C20 | IO,LDC | 7 |
| | U3 | X14 | IO,CS0 | 8 |
| GGG1 | ST2 | B23 | 23 | 7 |
| | U2 | D20 | IO | 7 |
| GGG2 | ST2 | B26 | 26 | 7 |
| | U2 | G20 | IO | 7 |
| GGG3 | ST2 | C28 | 28 | 7 |
| | U2 | H20 | IO | 7 |
| GGG4 | ST2 | A29 | 29 | 7 |
| | U2 | J20 | IO | 7 |
| GGG5 | J2 | 6 | TRIG | 2 |
| | J3 | 6 | TRIG | 2 |
| | U2 | M20 | IO | 7 |
| | U3 | J20 | IO | 8 |
| GGG6 | U2 | N20 | IO | 7 |
| | U3 | H20 | IO | 8 |
| GGG7 | U2 | P20 | IO | 7 |
| | U3 | G20 | IO | 8 |
| GGG8 | U2 | U20 | IO | 7 |
| | U3 | D20 | IO | 8 |
| GGG9 | U2 | V20 | IO | 7 |
| | U3 | C20 | IO,LDC | 8 |
| GGG11 | U2 | C19 | IO,HDC | 7 |
| | U3 | E17 | IO | 8 |
| GGG12 | ST2 | C23 | 23 | 7 |
| | U2 | D19 | IO | 7 |
| GGG13 | ST2 | A22 | 22 | 7 |
| | U2 | E19 | IO | 7 |
| GGG14 | ST2 | A24 | 24 | 7 |
| | U2 | F19 | IO | 7 |

| | | | | |
|---|---|---|---|---|
| GGG15 | ST2 | C26 | 26 | 7 |
| | U2 | G19 | IO | 7 |
| GGG16 | ST2 | A27 | 27 | 7 |
| | U2 | H19 | IO | 7 |
| GGG17 | ST2 | B29 | 29 | 7 |
| | U2 | J19 | IO | 7 |
| GGG19 | U2 | L19 | IO | 7 |
| | U3 | L19 | IO | 8 |
| GGG20 | U2 | M19 | IO | 7 |
| | U3 | J19 | IO | 8 |
| GGG21 | U2 | N19 | IO | 7 |
| | U3 | H19 | IO | 8 |
| GGG22 | U2 | P19 | IO | 7 |
| | U3 | G19 | IO | 8 |
| GGG23 | U2 | R19 | IO | 7 |
| | U3 | F19 | IO | 8 |
| GGG24 | U2 | T19 | IO | 7 |
| | U3 | E19 | IO | 8 |
| GGG25 | U2 | U19 | IO | 7 |
| | U3 | D19 | IO | 8 |
| GGG26 | U2 | V19 | IO | 7 |
| | U3 | C19 | IO,HDC | 8 |
| GGG27 | U2 | W19 | IO,D7 | 7 |
| | U3 | D18 | IO | 8 |
| GGG28 | ST2 | A21 | 21 | 7 |
| | U2 | D18 | IO | 7 |
| GGG29 | ST2 | B21 | 21 | 7 |
| | U2 | E18 | IO | 7 |
| GGG30 | ST2 | B24 | 24 | 7 |
| | U2 | F18 | IO | 7 |
| GGG31 | ST2 | A25 | 25 | 7 |
| | U2 | G18 | IO | 7 |
| GGG32 | ST2 | B27 | 27 | 7 |
| | U2 | H18 | IO | 7 |
| GGG33 | ST2 | C29 | 29 | 7 |
| | U2 | J18 | IO | 7 |
| GGG34 | ST2 | A30 | 30 | 7 |
| | U2 | K18 | IO | 7 |
| GGG35 | U2 | L18 | IO | 7 |
| | U3 | L18 | IO | 8 |
| GGG36 | U2 | M18 | IO | 7 |
| | U3 | K18 | IO | 8 |
| GGG37 | U2 | N18 | IO | 7 |
| | U3 | J18 | IO | 8 |
| GGG38 | U2 | P18 | IO | 7 |
| | U3 | H18 | IO | 8 |
| GGG39 | U2 | R18 | IO | 7 |
| | U3 | G18 | IO | 8 |
| GGG40 | U2 | T18 | IO | 7 |
| | U3 | F18 | IO | 8 |
| GGG41 | U2 | U18 | IO | 7 |
| | U3 | E18 | IO | 8 |
| GGG42 | ST2 | B20 | 20 | 7 |
| | U2 | E17 | IO | 7 |
| GGG43 | ST2 | C24 | 24 | 7 |
| | U2 | F17 | IO | 7 |
| GGG44 | ST2 | B25 | 25 | 7 |
| | U2 | G17 | IO | 7 |
| GGG45 | ST2 | C27 | 27 | 7 |
| | U2 | H17 | IO | 7 |
| GGG46 | ST2 | A28 | 28 | 7 |
| | U2 | J17 | IO | 7 |
| GGG47 | ST2 | B30 | 30 | 7 |
| | U2 | K17 | IO | 7 |
| GGG48 | U2 | L17 | IO | 7 |
| | U3 | L17 | IO | 8 |
| GGG49 | U2 | M17 | IO | 7 |
| | U3 | K17 | IO | 8 |
| GGG50 | U2 | N17 | IO | 7 |
| | U3 | J17 | IO | 8 |
| GGG51 | U2 | P17 | IO | 7 |
| | U3 | H17 | IO | 8 |
| GGG52 | U2 | R17 | IO | 7 |
| | U3 | G17 | IO | 8 |
| GGG53 | U2 | T17 | IO | 7 |
| | U3 | F17 | IO | 8 |
| GGG54 | ST2 | A23 | 23 | 7 |
| | U2 | F16 | IO | 7 |
| GGG55 | ST2 | C25 | 25 | 7 |
| | U2 | G16 | IO | 7 |
| GGG56 | ST2 | A26 | 26 | 7 |
| | U2 | H16 | IO | 7 |
| GGG57 | ST2 | B28 | 28 | 7 |
| | U2 | J16 | IO | 7 |
| GGG58 | ST2 | C30 | 30 | 7 |
| | U2 | K16 | IO | 7 |
| GGG59 | U2 | L16 | IO | 7 |
| | U3 | K16 | IO | 8 |
| GGG60 | U2 | M16 | IO | 7 |
| | U3 | J16 | IO | 8 |
| GGG61 | U2 | N16 | IO | 7 |
| | U3 | H16 | IO | 8 |
| GGG62 | U2 | P16 | IO | 7 |
| | U3 | G16 | IO | 8 |
| GGG63 | U2 | R16 | IO | 7 |
| | U3 | F16 | IO | 8 |
| GND | B1 | GND | GND | 3 |
| | C1 | 1 | 1 | 15 |
| | C10 | 1 | 1 | 15 |
| | C11 | 1 | 1 | 15 |
| | C12 | 1 | 1 | 15 |
| | C13 | 1 | 1 | 15 |
| | C14 | 1 | 1 | 15 |
| | C15 | 1 | 1 | 15 |
| | C16 | 1 | 1 | 15 |
| | C17 | 1 | 1 | 15 |
| | C18 | 1 | 1 | 15 |
| | C19 | 1 | 1 | 15 |
| | C2 | 1 | 1 | 15 |
| | C20 | 1 | 1 | 15 |
| | C21 | 1 | 1 | 15 |
| | C22 | 1 | 1 | 15 |
| | C23 | 1 | 1 | 15 |
| | C24 | 1 | 1 | 15 |
| | C25 | 1 | 1 | 15 |
| | C26 | 1 | 1 | 15 |
| | C27 | 1 | 1 | 15 |
| | C28 | 1 | 1 | 15 |
| | C29 | 1 | 1 | 15 |
| | C3 | 1 | 1 | 15 |
| | C30 | 1 | 1 | 15 |
| | C31 | 1 | 1 | 15 |
| | C32 | 1 | 1 | 15 |
| | C33 | 1 | 1 | 15 |
| | C34 | 1 | 1 | 15 |
| | C35 | 1 | 1 | 15 |
| | C36 | 1 | 1 | 15 |
| | C37 | 1 | 1 | 15 |
| | C38 | 1 | 1 | 15 |
| | C39 | 1 | 1 | 15 |
| | C4 | 1 | 1 | 15 |
| | C40 | 1 | 1 | 15 |
| | C41 | 1 | 1 | 15 |
| | C42 | 1 | 1 | 15 |
| | C43 | 1 | 1 | 15 |
| | C44 | 1 | 1 | 15 |
| | C45 | 1 | 1 | 15 |
| | C46 | 1 | 1 | 15 |
| | C47 | 1 | 1 | 15 |
| | C48 | 1 | 1 | 15 |
| | C49 | 1 | 1 | 15 |
| | C5 | 1 | 1 | 15 |
| | C50 | 1 | 1 | 15 |
| | C51 | 1 | 1 | 15 |
| | C52 | 1 | 1 | 15 |
| | C53 | 1 | 1 | 15 |
| | C54 | 1 | 1 | 15 |
| | C55 | 1 | 1 | 15 |
| | C56 | 1 | 1 | 15 |
| | C57 | 1 | 1 | 15 |
| | C58 | 1 | 1 | 15 |
| | C59 | 1 | 1 | 15 |
| | C6 | 1 | 1 | 15 |
| | C60 | 1 | 1 | 15 |
| | C61 | 1 | 1 | 15 |
| | C62 | 1 | 1 | 15 |
| | C63 | 1 | 1 | 15 |
| | C64 | 1 | 1 | 15 |
| | C65 | 1 | 1 | 15 |
| | C66 | 1 | 1 | 15 |
| | C67 | 1 | 1 | 15 |
| | C68 | 1 | 1 | 15 |
| | C69 | 1 | 1 | 15 |
| | C7 | 1 | 1 | 15 |
| | C70 | 1 | 1 | 15 |
| | C71 | 1 | 1 | 15 |
| | C72 | 1 | 1 | 15 |
| | C73 | 1 | 1 | 15 |
| | C74 | 1 | 1 | 15 |
| | C75 | 1 | 1 | 15 |
| | C76 | 1 | 1 | 15 |
| | C77 | 1 | 1 | 15 |
| | C78 | 1 | 1 | 15 |
| | C79 | 1 | 1 | 1 |
| | C8 | 1 | 1 | 15 |
| | C80 | 1 | 1 | 15 |
| | C81 | 1 | 1 | 15 |
| | C82 | 1 | 1 | 15 |
| | C83 | 1 | 1 | 3 |
| | C84 | 1 | 1 | 3 |
| | C85 | 1 | 1 | 1 |
| | C86 | 1 | 1 | 1 |
| | C87 | 1 | 1 | 1 |
| | C88 | 1 | 1 | 3 |
| | C89 | 1 | 1 | 15 |
| | C9 | 1 | 1 | 15 |
| | C90 | 1 | 1 | 15 |
| | C91 | 1 | 1 | 15 |
| | C92 | 1 | 1 | 3 |
| | C93 | 1 | 1 | 3 |
| | C94 | 1 | 1 | 3 |
| | C95 | 2 | 2 | 6 |
| | C96 | 2 | 2 | 6 |
| | IC1 | 4 | GND | 3 |
| | IC2 | 12 | D7 | 1 |
| | IC2 | 13 | D6 | 1 |
| | IC2 | 14 | D5 | 1 |
| | IC2 | 15 | D4 | 1 |
| | IC2 | 7 | G | 1 |
| | IC2 | 8 | GND | 1 |
| | IC2 | 9 | C | 1 |
| | IC3 | 1 | D3 | 1 |
| | IC3 | 2 | D2 | 1 |
| | IC3 | 3 | D1 | 1 |
| | IC3 | 7 | G | 1 |
| | IC3 | 8 | GND | 1 |
| | IC4 | 7 | GND | 3 |
| | IC5 | 1 | GND | 6 |
| | IC6 | 12 | OE_B# | 1 |
| | IC6 | 16 | GND | 1 |
| | IC6 | 5 | GND | 1 |
| | IC6 | 8 | GND | 1 |
| | IC6 | 9 | OE_A# | 1 |
| | IC7 | 12 | OE_B# | 1 |
| | IC7 | 16 | GND | 1 |
| | IC7 | 5 | GND | 1 |
| | IC7 | 8 | GND | 1 |
| | IC7 | 9 | OE_A# | 1 |
| | J1 | 3 | GND | 2 |
| | J2 | 3 | GND | 2 |
| | J3 | 3 | GND | 2 |
| | J4 | 3 | GND | 2 |
| | J5 | 2 | SHIELD | 1 |
| | JP14 | 3 | 3 | 1 |
| | JP15 | 3 | 3 | 1 |
| | JP16 | 3 | 3 | 1 |
| | JP17 | 3 | 3 | 1 |
| | JP18 | 3 | 3 | 6 |
| | JP19 | 2 | 2 | 6 |
| | JP20 | 3 | 3 | 6 |
| | JP21 | 32 | 32 | 3 |
| | JP22 | 3 | 3 | 6 |
| | JP24 | 3 | 3 | 7 |
| | JP26 | 3 | 3 | 7 |
| | JP27 | 3 | 3 | 6 |
| | JP28 | 3 | 3 | 6 |
| | JP30 | 3 | 3 | 1 |
| | JP35 | 1 | 1 | 2 |
| | JP36 | 1 | 1 | 2 |
| | JP37 | 1 | 1 | 2 |

| | | | | |
|---|---|---|---|---|
| | JP9 | 3 | 3 | 6 |
| | QG1 | 4 | GND | 1 |
| | R8 | 1 | 1 | 3 |
| | S1 | 3 | 3 | 2 |
| | S1 | 4 | 4 | 2 |
| | S2 | 3 | 3 | 2 |
| | S2 | 4 | 4 | 2 |
| | ST1 | A15 | 15 | 6 |
| | ST1 | A19 | 19 | 6 |
| | ST1 | C9 | 9 | 6 |
| | ST2 | A15 | 15 | 7 |
| | ST2 | A19 | 19 | 7 |
| | ST2 | B2 | 2 | 7 |
| | ST2 | B31 | 31 | 7 |
| | ST2 | C9 | 9 | 7 |
| | ST3 | A14 | 14 | 8 |
| | ST3 | A18 | 18 | 8 |
| | ST3 | B2 | 2 | 8 |
| | ST3 | B31 | 31 | 8 |
| | ST3 | C24 | 24 | 8 |
| | ST4 | A14 | 14 | 9 |
| | ST4 | A18 | 18 | 9 |
| | ST4 | B2 | 2 | 9 |
| | ST4 | B31 | 31 | 9 |
| | ST4 | C24 | 24 | 9 |
| | U1 | A10 | GND | 11 |
| | U1 | A15 | GND | 11 |
| | U1 | A19 | GND | 11 |
| | U1 | A5 | GND | 11 |
| | U1 | B1 | GND | 11 |
| | U1 | E16 | GND | 11 |
| | U1 | E20 | GND | 11 |
| | U1 | F1 | GND | 11 |
| | U1 | K20 | GND | 11 |
| | U1 | L1 | GND | 11 |
| | U1 | R20 | GND | 11 |
| | U1 | T1 | GND | 11 |
| | U1 | T5 | GND | 11 |
| | U1 | W20 | GND | 11 |
| | U1 | X11 | GND | 11 |
| | U1 | X16 | GND | 11 |
| | U1 | X2 | GND | 11 |
| | U1 | X6 | GND | 11 |
| | U10 | 100 | GND | 5 |
| | U10 | 110 | GND | 5 |
| | U10 | 111 | GND | 5 |
| | U10 | 114 | GND | 5 |
| | U10 | 19 | GND | 5 |
| | U10 | 53 | GND | 5 |
| | U10 | 56 | GND | 5 |
| | U10 | 57 | GND | 5 |
| | U10 | 63 | GND | 5 |
| | U10 | 77 | GND | 5 |
| | U10 | 80 | GND | 5 |
| | U10 | 84 | GND | 5 |
| | U10 | 90 | GND | 5 |
| | U2 | A10 | GND | 12 |
| | U2 | A15 | GND | 12 |
| | U2 | A19 | GND | 12 |
| | U2 | A5 | GND | 12 |
| | U2 | B1 | GND | 12 |
| | U2 | E16 | GND | 12 |
| | U2 | E20 | GND | 12 |
| | U2 | F1 | GND | 12 |
| | U2 | K20 | GND | 12 |
| | U2 | L1 | GND | 12 |
| | U2 | R20 | GND | 12 |
| | U2 | T1 | GND | 12 |
| | U2 | T5 | GND | 12 |
| | U2 | W20 | GND | 12 |
| | U2 | X11 | GND | 12 |
| | U2 | X16 | GND | 12 |
| | U2 | X2 | GND | 12 |
| | U2 | X6 | GND | 12 |
| | U3 | A10 | GND | 12 |
| | U3 | A15 | GND | 12 |
| | U3 | A19 | GND | 12 |
| | U3 | A5 | GND | 12 |
| | U3 | B1 | GND | 12 |
| | U3 | E16 | GND | 12 |
| | U3 | E20 | GND | 12 |
| | U3 | F1 | GND | 12 |
| | U3 | K20 | GND | 12 |
| | U3 | L1 | GND | 12 |
| | U3 | R20 | GND | 12 |
| | U3 | T1 | GND | 12 |
| | U3 | T5 | GND | 12 |
| | U3 | W20 | GND | 12 |
| | U3 | X11 | GND | 12 |
| | U3 | X16 | GND | 12 |
| | U3 | X2 | GND | 12 |
| | U3 | X6 | GND | 12 |
| | U4 | A10 | GND | 12 |
| | U4 | A15 | GND | 12 |
| | U4 | A19 | GND | 12 |
| | U4 | A5 | GND | 12 |
| | U4 | B1 | GND | 12 |
| | U4 | E16 | GND | 12 |
| | U4 | E20 | GND | 12 |
| | U4 | F1 | GND | 12 |
| | U4 | K20 | GND | 12 |
| | U4 | L1 | GND | 12 |
| | U4 | R20 | GND | 12 |
| | U4 | T1 | GND | 12 |
| | U4 | T5 | GND | 12 |
| | U4 | W20 | GND | 12 |
| | U4 | X11 | GND | 12 |
| | U4 | X16 | GND | 12 |
| | U4 | X2 | GND | 12 |
| | U4 | X6 | GND | 12 |
| | U5 | P10 | VSS | 4 |
| | U5 | P14 | VSS | 4 |
| | U5 | P17 | VSS | 4 |
| | U5 | P21 | VSS | 4 |
| | U5 | P26 | VSS | 4 |
| | U5 | P40 | VSS | 4 |
| | U5 | P5 | VSS | 4 |
| | U5 | P55 | VSS | 4 |
| | U5 | P60 | VSS | 4 |
| | U5 | P64 | VSS | 4 |
| | U5 | P66 | VSS | 4 |
| | U5 | P67 | VSS | 4 |
| | U5 | P71 | VSS | 4 |
| | U5 | P76 | VSS | 4 |
| | U5 | P90 | VSS | 4 |
| | U5 | P93 | BW1# | 4 |
| | U5 | P94 | BW2# | 4 |
| | U5 | P95 | BW3# | 4 |
| | U5 | P96 | BW4# | 4 |
| | U5 | P98 | CE1# | 4 |
| | U6 | P10 | VSS | 4 |
| | U6 | P14 | VSS | 4 |
| | U6 | P17 | VSS | 4 |
| | U6 | P21 | VSS | 4 |
| | U6 | P26 | VSS | 4 |
| | U6 | P40 | VSS | 4 |
| | U6 | P5 | VSS | 4 |
| | U6 | P55 | VSS | 4 |
| | U6 | P60 | VSS | 4 |
| | U6 | P64 | VSS | 4 |
| | U6 | P66 | VSS | 4 |
| | U6 | P67 | VSS | 4 |
| | U6 | P71 | VSS | 4 |
| | U6 | P76 | VSS | 4 |
| | U6 | P90 | VSS | 4 |
| | U6 | P93 | BW1# | 4 |
| | U6 | P94 | BW2# | 4 |
| | U6 | P95 | BW3# | 4 |
| | U6 | P96 | BW4# | 4 |
| | U7 | P16 | VSS | 3 |
| | U8 | P16 | VSS | 3 |
| | U9 | 100 | GND | 5 |
| | U9 | 110 | GND | 5 |
| | U9 | 111 | GND | 5 |
| | U9 | 114 | GND | 5 |
| | U9 | 19 | GND | 5 |
| | U9 | 53 | GND | 5 |
| | U9 | 56 | GND | 5 |
| | U9 | 57 | GND | 5 |
| | U9 | 63 | GND | 5 |
| | U9 | 77 | GND | 5 |
| | U9 | 80 | GND | 5 |
| | U9 | 84 | GND | 5 |
| | U9 | 90 | GND | 5 |
| | X1 | 2 | KL | 6 |
| GTS | JP21 | 36 | 36 | 3 |
| | U1 | A3 | IO | 10 |
| | U2 | X7 | IO | 12 |
| | U3 | A7 | IO | 12 |
| | U4 | X3 | IO | 10 |
| HHH0 | JP23 | 1 | 1 | 7 |
| | U2 | A3 | IO | 7 |
| HHH1 | ST2 | C2 | 2 | 7 |
| | U2 | A4 | IO | 7 |
| HHH2 | ST2 | A5 | 5 | 7 |
| | U2 | A7 | IO | 7 |
| HHH3 | ST2 | B7 | 7 | 7 |
| | U2 | A8 | IO | 7 |
| HHH4 | ST2 | B9 | 9 | 7 |
| | U2 | A9 | IO | 7 |
| HHH5 | ST2 | A13 | 13 | 7 |
| | U2 | A12 | IO | 7 |
| HHH6 | ST2 | B15 | 15 | 7 |
| | U2 | A13 | IO | 7 |
| HHH7 | ST2 | B17 | 17 | 7 |
| | U2 | A14 | IO | 7 |
| HHH8 | ST2 | C21 | 21 | 7 |
| | U2 | A17 | IO | 7 |
| HHH9 | JP26 | 1 | 1 | 7 |
| | U2 | A18 | IO | 7 |
| HHH10 | ST2 | C1 | 1 | 7 |
| | U2 | B3 | IO | 7 |
| HHH11 | ST2 | A1 | 1 | 7 |
| | U2 | B4 | IO | 7 |
| HHH12 | ST2 | C3 | 3 | 7 |
| | U2 | B5 | IO | 7 |
| HHH13 | ST2 | B4 | 4 | 7 |
| | U2 | B6 | IO | 7 |
| HHH14 | ST2 | B5 | 5 | 7 |
| | U2 | B7 | IO | 7 |
| HHH15 | ST2 | C7 | 7 | 7 |
| | U2 | B8 | IO | 7 |
| HHH16 | ST2 | A8 | 8 | 7 |
| | U2 | B9 | IO | 7 |
| HHH18 | ST2 | C12 | 12 | 7 |
| | U2 | B11 | IO | 7 |
| HHH19 | JP25 | 1 | 1 | 7 |
| | U2 | B12 | IO | 7 |
| HHH20 | ST2 | C15 | 15 | 7 |
| | U2 | B13 | IO | 7 |
| HHH21 | ST2 | C17 | 17 | 7 |
| | U2 | B14 | IO | 7 |
| HHH22 | ST2 | A18 | 18 | 7 |
| | U2 | B15 | IO | 7 |
| HHH23 | ST2 | C20 | 20 | 7 |
| | U2 | B16 | IO | 7 |
| HHH24 | ST2 | A20 | 20 | 7 |
| | U2 | B17 | IO | 7 |
| HHH25 | ST2 | C22 | 22 | 7 |
| | U2 | B18 | IO | 7 |
| HHH28 | ST2 | A2 | 2 | 7 |
| | U2 | C5 | IO | 7 |
| HHH29 | ST2 | C4 | 4 | 7 |
| | U2 | C6 | IO | 7 |
| HHH31 | ST2 | A6 | 6 | 7 |
| | U2 | C8 | IO | 7 |
| HHH32 | ST2 | B8 | 8 | 7 |
| | U2 | C9 | IO | 7 |
| HHH33 | ST2 | B10 | 10 | 7 |
| | U2 | C10 | IO | 7 |
| HHH34 | ST2 | A11 | 11 | 7 |
| | U2 | C11 | IO | 7 |
| HHH35 | ST2 | C13 | 13 | 7 |
| | U2 | C12 | IO | 7 |
| HHH36 | ST2 | A14 | 14 | 7 |
| | U2 | C13 | IO | 7 |
| HHH37 | ST2 | A16 | 16 | 7 |
| | U2 | C14 | IO | 7 |
| HHH38 | ST2 | B18 | 18 | 7 |
| | U2 | C15 | IO | 7 |
| HHH39 | ST2 | B19 | 19 | 7 |
| | U2 | C16 | IO | 7 |
| HHH42 | ST2 | A3 | 3 | 7 |
| | U2 | D6 | IO | 7 |
| HHH43 | ST2 | C5 | 5 | 7 |
| | U2 | D7 | IO | 7 |
| HHH44 | ST2 | B6 | 6 | 7 |
| | U2 | D8 | IO | 7 |

| | | | | |
|---|---|---|---|---|
| HHH45 | ST2 | C8 | 8 | 7 |
| | U2 | D9 | IO | 7 |
| HHH46 | ST2 | C10 | 10 | 7 |
| | U2 | D10 | IO | 7 |
| HHH47 | ST2 | B11 | 11 | 7 |
| | U2 | D11 | IO | 7 |
| HHH48 | ST2 | A12 | 12 | 7 |
| | U2 | D12 | IO | 7 |
| HHH49 | ST2 | B14 | 14 | 7 |
| | U2 | D13 | IO | 7 |
| HHH50 | ST2 | B16 | 16 | 7 |
| | U2 | D14 | IO | 7 |
| HHH51 | ST2 | C18 | 18 | 7 |
| | U2 | D15 | IO | 7 |
| HHH52 | ST2 | C19 | 19 | 7 |
| | U2 | D16 | IO | 7 |
| HHH53 | ST2 | B3 | 3 | 7 |
| | U2 | E6 | IO | 7 |
| HHH54 | ST2 | A4 | 4 | 7 |
| | U2 | E7 | IO | 7 |
| HHH55 | ST2 | C6 | 6 | 7 |
| | U2 | E8 | IO | 7 |
| HHH56 | ST2 | A7 | 7 | 7 |
| | U2 | E9 | IO | 7 |
| HHH57 | ST2 | A9 | 9 | 7 |
| | U2 | E10 | IO | 7 |
| HHH58 | ST2 | C11 | 11 | 7 |
| | U2 | E11 | IO | 7 |
| HHH59 | JP24 | 1 | 1 | 7 |
| | U2 | E12 | IO | 7 |
| HHH60 | ST2 | C14 | 14 | 7 |
| | U2 | E13 | IO | 7 |
| HHH61 | ST2 | C16 | 16 | 7 |
| | U2 | E14 | IO | 7 |
| HHH62 | ST2 | A17 | 17 | 7 |
| | U2 | E15 | IO | 7 |
| INI1 | J1 | 15 | INIT | 2 |
| | JP7 | 1 | 1 | 6 |
| | R41 | 1 | 1 | 2 |
| | U1 | K19 | IO,INIT | 6 |
| INI2 | J2 | 15 | INIT | 2 |
| | JP7 | 2 | 2 | 6 |
| | JP8 | 1 | 1 | 7 |
| | U2 | K19 | IO,INIT | 7 |
| INI3 | J3 | 15 | INIT | 2 |
| | JP29 | 1 | 1 | 8 |
| | JP8 | 2 | 2 | 7 |
| | U3 | K19 | IO,INIT | 8 |
| INI4 | J4 | 15 | INIT | 2 |
| | JP29 | 2 | 2 | 8 |
| | U4 | K19 | IO,INIT | 14 |
| JJJ1 | ST3 | B32 | 32 | 8 |
| | U3 | X3 | IO | 8 |
| JJJ2 | ST3 | C31 | 31 | 8 |
| | U3 | X4 | IO | 8 |
| JJJ3 | ST3 | C27 | 27 | 8 |
| | U3 | X7 | IO | 8 |
| JJJ4 | ST3 | A26 | 26 | 8 |
| | U3 | X8 | IO | 8 |
| JJJ5 | ST3 | A24 | 24 | 8 |
| | U3 | X9 | IO | 8 |
| JJJ6 | ST3 | A20 | 20 | 8 |
| | U3 | X12 | IO | 8 |
| JJJ7 | ST3 | B18 | 18 | 8 |
| | U3 | X13 | IO | 8 |
| JJJ9 | ST3 | A13 | 13 | 8 |
| | U3 | X17 | IO | 8 |
| JJJ10 | ST3 | C12 | 12 | 8 |
| | U3 | X18 | IO | 8 |
| JJJ11 | U2 | W2 | IO,A0,WS | 12 |
| | U3 | W2 | IO,A0,WS | 8 |
| JJJ12 | ST3 | C32 | 32 | 8 |
| | U3 | W3 | IO | 8 |
| JJJ13 | ST3 | A32 | 32 | 8 |
| | U3 | W4 | IO | 8 |
| JJJ14 | ST3 | C30 | 30 | 8 |
| | U3 | W5 | IO | 8 |
| JJJ15 | ST3 | B29 | 29 | 8 |
| | U3 | W6 | IO | 8 |
| JJJ16 | ST3 | A28 | 28 | 8 |
| | U3 | W7 | IO | 8 |
| JJJ17 | ST3 | B26 | 26 | 8 |
| | U3 | W8 | IO,D2 | 8 |
| JJJ18 | ST3 | B24 | 24 | 8 |
| | U3 | W9 | IO | 8 |
| JJJ19 | U2 | V7 | IO | 12 |
| | U3 | W10 | IO,D3 | 8 |
| JJJ20 | ST3 | C21 | 21 | 8 |
| | U3 | W11 | IO | 8 |
| JJJ21 | ST3 | B20 | 20 | 8 |
| | U3 | W12 | IO | 8 |
| JJJ22 | ST3 | C18 | 18 | 8 |
| | U3 | W13 | IO | 8 |
| JJJ23 | ST3 | C16 | 16 | 8 |
| | U3 | W14 | IO | 8 |
| JJJ24 | ST3 | B15 | 15 | 8 |
| | U3 | W15 | IO | 8 |
| JJJ25 | ST3 | B14 | 14 | 8 |
| | U3 | W16 | IO | 8 |
| JJJ26 | ST3 | B13 | 13 | 8 |
| | U3 | W17 | IO,D6 | 8 |
| JJJ29 | U2 | U5 | IO | 12 |
| | U3 | V5 | IO,RCLK,RB | 8 |
| JJJ30 | ST3 | C29 | 29 | 8 |
| | U3 | V6 | IO | 8 |
| JJJ31 | ST3 | B28 | 28 | 8 |
| | U3 | V7 | IO | 8 |
| JJJ32 | ST3 | C26 | 26 | 8 |
| | U3 | V8 | IO | 8 |
| JJJ33 | ST3 | A25 | 25 | 8 |
| | U3 | V9 | IO | 8 |
| JJJ34 | JP41 | 1 | 1 | 8 |
| | R39 | 1 | 1 | 8 |
| | U3 | V10 | IO,RS | 8 |
| JJJ35 | ST3 | A22 | 22 | 8 |
| | U3 | V11 | IO,D4 | 8 |
| JJJ36 | ST3 | C20 | 20 | 8 |
| | U3 | V12 | IO | 8 |
| JJJ37 | ST3 | A19 | 19 | 8 |
| | U3 | V13 | IO | 8 |
| JJJ38 | ST3 | A17 | 17 | 8 |
| | U3 | V14 | IO | 8 |
| JJJ39 | ST3 | C15 | 15 | 8 |
| | U3 | V15 | IO | 8 |
| JJJ40 | ST3 | C14 | 14 | 8 |
| | U3 | V16 | IO | 8 |
| JJJ41 | ST3 | C13 | 13 | 8 |
| | U3 | V17 | IO | 8 |
| JJJ43 | ST3 | A31 | 31 | 8 |
| | U3 | U5 | IO | 8 |
| JJJ44 | ST3 | A30 | 30 | 8 |
| | U3 | U6 | IO,D1 | 8 |
| JJJ45 | ST3 | C28 | 28 | 8 |
| | U3 | U7 | IO | 8 |
| JJJ46 | ST3 | A27 | 27 | 8 |
| | U3 | U8 | IO | 8 |
| JJJ47 | ST3 | B25 | 25 | 8 |
| | U3 | U9 | IO | 8 |
| JJJ48 | ST3 | B23 | 23 | 8 |
| | U3 | U10 | IO | 8 |
| JJJ49 | ST3 | B22 | 22 | 8 |
| | U3 | U11 | IO | 8 |
| JJJ50 | ST3 | A21 | 21 | 8 |
| | U3 | U12 | IO | 8 |
| JJJ51 | ST3 | B19 | 19 | 8 |
| | U3 | U13 | IO | 8 |
| JJJ52 | ST3 | B17 | 17 | 8 |
| | U3 | U14 | IO | 8 |
| JJJ53 | ST3 | A16 | 16 | 8 |
| | U3 | U15 | IO | 8 |
| JJJ54 | ST3 | A15 | 15 | 8 |
| | U3 | U16 | IO | 8 |
| JJJ55 | ST3 | B30 | 30 | 8 |
| | U3 | T6 | IO | 8 |
| JJJ56 | ST3 | A29 | 29 | 8 |
| | U3 | T7 | IO | 8 |
| JJJ57 | ST3 | B27 | 27 | 8 |
| | U3 | T8 | IO | 8 |
| JJJ58 | ST3 | C25 | 25 | 8 |
| | U3 | T9 | IO | 8 |
| JJJ59 | ST3 | C23 | 23 | 8 |
| | U3 | T10 | IO | 8 |
| JJJ60 | ST3 | C22 | 22 | 8 |
| | U3 | T11 | IO | 8 |
| JJJ61 | ST3 | B21 | 21 | 8 |
| | U3 | T12 | IO,D5 | 8 |
| JJJ62 | ST3 | C19 | 19 | 8 |
| | U3 | T13 | IO | 8 |
| JJJ63 | ST3 | C17 | 17 | 8 |
| | U3 | T14 | IO | 8 |
| JJJ64 | ST3 | B16 | 16 | 8 |
| | U3 | T15 | IO | 8 |
| KKK5 | ST3 | B3 | 3 | 8 |
| | U3 | H20 | IO | 8 |
| KKK6 | ST3 | A5 | 5 | 8 |
| | U3 | N20 | IO | 8 |
| KKK7 | ST3 | C6 | 6 | 8 |
| | U3 | P20 | IO | 8 |
| KKK8 | ST3 | C10 | 10 | 8 |
| | U3 | U20 | IO | 8 |
| KKK9 | ST3 | C11 | 11 | 8 |
| | U3 | V20 | IO | 8 |
| KKK20 | ST3 | C3 | 3 | 8 |
| | U3 | H19 | IO | 8 |
| KKK21 | ST3 | B5 | 5 | 8 |
| | U3 | N19 | IO | 8 |
| KKK22 | ST3 | A7 | 7 | 8 |
| | U3 | P19 | IO | 8 |
| KKK23 | ST3 | B8 | 8 | 8 |
| | U3 | R19 | IO | 8 |
| KKK24 | ST3 | C9 | 9 | 8 |
| | U3 | T19 | IO | 8 |
| KKK25 | ST3 | A11 | 11 | 8 |
| | U3 | U19 | IO | 8 |

| | | | | |
|---|---|---|---|---|
| KKK26 | ST3 | A12 | 12 | 8 |
| | U3 | V19 | IO | 8 |
| KKK27 | ST3 | B12 | 12 | 8 |
| | U3 | W19 | IO,D7 | 8 |
| KKK36 | ST3 | A4 | 4 | 8 |
| | U3 | H18 | IO | 8 |
| KKK37 | ST3 | C5 | 5 | 8 |
| | U3 | N18 | IO | 8 |
| KKK38 | ST3 | B7 | 7 | 8 |
| | U3 | P18 | IO | 8 |
| KKK39 | ST3 | C8 | 8 | 8 |
| | U3 | R18 | IO | 8 |
| KKK40 | ST3 | A10 | 10 | 8 |
| | U3 | T18 | IO | 8 |
| KKK41 | ST3 | B11 | 11 | 8 |
| | U3 | U18 | IO | 8 |
| KKK49 | ST3 | B4 | 4 | 8 |
| | U3 | H17 | IO | 8 |
| KKK50 | ST3 | A6 | 6 | 8 |
| | U3 | N17 | IO | 8 |
| KKK51 | ST3 | C7 | 7 | 8 |
| | U3 | P17 | IO | 8 |
| KKK52 | ST3 | A9 | 9 | 8 |
| | U3 | R17 | IO | 8 |
| KKK53 | ST3 | B10 | 10 | 8 |
| | U3 | T17 | IO | 8 |
| KKK59 | ST3 | A3 | 3 | 8 |
| | U3 | L16 | IO | 8 |
| KKK60 | ST3 | C4 | 4 | 8 |
| | U3 | H16 | IO | 8 |
| KKK61 | ST3 | B6 | 6 | 8 |
| | U3 | N16 | IO | 8 |
| KKK62 | ST3 | A8 | 8 | 8 |
| | U3 | P16 | IO | 8 |
| KKK63 | ST3 | B9 | 9 | 8 |
| | U3 | R16 | IO | 8 |
| LLL0 | U3 | C1 | IO,A13 | 12 |
| | U4 | V1 | IO,A3 | 12 |
| LLL1 | U3 | D1 | IO | 12 |
| | U4 | U1 | IO | 12 |
| LLL2 | U10 | 12 | A2_R | 5 |
| | U3 | G1 | IO | 12 |
| | U9 | 12 | A2_R | 5 |
| LLL3 | U10 | 126 | A12_R | 5 |
| | U3 | H1 | IO,A10 | 12 |
| | U9 | 126 | A12_R | 5 |
| LLL4 | U10 | 116 | CNTRST#_R | 5 |
| | U3 | J1 | IO | 12 |
| | U9 | 116 | CNTRST#_R | 5 |
| LLL5 | U3 | M1 | IO,A21 | 12 |
| | U9 | 85 | IO0_R | 5 |
| LLL6 | U10 | 91 | IO4_R | 5 |
| | U3 | N1 | IO,A4 | 12 |
| LLL7 | U3 | P1 | IO | 12 |
| | U9 | 102 | IO10_R | 5 |
| LLL8 | U3 | U1 | IO | 12 |
| | U9 | 101 | IO9_R | 5 |
| LLL9 | U10 | 101 | IO9_R | 5 |
| | U3 | V1 | IO,A3 | 12 |
| LLL10 | U3 | B2 | IO,A17 | 12 |
| | U4 | C2 | IO | 12 |
| LLL11 | U3 | C2 | IO | 12 |
| | U4 | B2 | IO,A17 | 12 |
| LLL12 | U3 | D2 | IO | 12 |
| | U4 | U2 | IO | 12 |
| LLL13 | U10 | 13 | A1_R | 5 |
| | U3 | E2 | IO | 12 |
| | U9 | 13 | A1_R | 5 |
| LLL14 | U10 | 14 | A0_R | 5 |
| | U3 | F2 | IO | 12 |
| | U9 | 14 | A0_R | 5 |
| LLL15 | U10 | 16 | CNTEN#_R | 5 |
| | U3 | G2 | IO | 12 |
| | U9 | 16 | CNTEN#_R | 5 |
| LLL16 | U10 | 127 | A11_R | 5 |
| | U3 | H2 | IO | 12 |
| | U9 | 127 | A11_R | 5 |
| LLL17 | U10 | 118 | CE0#_R | 5 |
| | U3 | J2 | IO | 12 |
| | U9 | 118 | CE0#_R | 5 |
| LLL18 | U3 | K2 | IO,A8 | 12 |
| | U9 | 109 | IO15_R | 5 |
| LLL19 | U3 | L2 | IO,A7 | 12 |
| | U9 | 107 | IO13_R | 5 |
| LLL20 | U3 | M2 | IO | 12 |
| | U9 | 103 | IO11_R | 5 |
| LLL21 | U3 | N2 | IO | 12 |
| | U9 | 91 | IO4_R | 5 |
| LLL22 | U10 | 98 | IO8_R | 5 |
| | U3 | P2 | IO | 12 |
| LLL23 | U3 | R2 | IO | 12 |
| | U9 | 89 | IO3_R | 5 |
| LLL24 | U10 | 92 | IO5_R | 5 |
| | U3 | T2 | IO | 12 |
| LLL25 | U10 | 95 | IO7_R | 5 |
| | U3 | U2 | IO | 12 |
| LLL28 | U3 | D3 | IO,A14 | 12 |
| | U4 | U3 | IO,A2,CS1 | 12 |
| LLL29 | U3 | E3 | IO | 12 |
| | U4 | T3 | IO | 12 |
| LLL30 | U10 | 18 | ADS#_R | 5 |
| | U3 | F3 | IO | 12 |
| | U9 | 18 | ADS#_R | 5 |
| LLL31 | U10 | 5 | A9_R | 5 |
| | U3 | G3 | IO | 12 |
| | U9 | 5 | A9_R | 5 |
| LLL32 | U10 | 128 | A10_R | 5 |
| | U3 | H3 | IO | 12 |
| | U9 | 128 | A10_R | 5 |
| LLL33 | U10 | 119 | UB#_R | 5 |
| | U10 | 120 | LB#_R | 5 |
| | U3 | J3 | IO,A11 | 12 |
| | U9 | 119 | UB#_R | 5 |
| | U9 | 120 | LB#_R | 5 |
| LLL34 | U10 | 109 | IO15_R | 5 |
| | U3 | K3 | IO,A9 | 12 |
| LLL35 | U10 | 107 | IO13_R | 5 |
| | U3 | L3 | IO,A6 | 12 |
| LLL36 | U10 | 103 | IO11_R | 5 |
| | U3 | M3 | IO,A5 | 12 |
| LLL37 | U10 | 87 | IO2_R | 5 |
| | U3 | N3 | IO | 12 |
| LLL38 | U3 | P3 | IO | 12 |
| | U9 | 98 | IO8_R | 5 |
| LLL39 | U10 | 86 | IO1_R | 5 |
| | U3 | R3 | IO | 12 |
| LLL40 | U3 | T3 | IO | 12 |
| | U9 | 92 | IO5_R | 5 |
| LLL41 | U3 | U3 | IO,A2,CS1 | 12 |
| | U9 | 95 | IO7_R | 5 |
| LLL43 | U3 | E4 | IO | 12 |
| | U4 | T4 | IO | 12 |
| LLL44 | U10 | 7 | A7_R | 5 |
| | U3 | F4 | IO | 12 |
| | U9 | 7 | A7_R | 5 |
| LLL45 | U10 | 8 | A6_R | 5 |
| | U3 | G4 | IO | 12 |
| | U9 | 8 | A6_R | 5 |
| LLL46 | U10 | 6 | A8_R | 5 |
| | U3 | H4 | IO | 12 |
| | U9 | 6 | A8_R | 5 |
| LLL47 | U10 | 124 | A14_R | 5 |
| | U3 | J4 | IO | 12 |
| | U9 | 124 | A14_R | 5 |
| LLL48 | U10 | 112 | OE#_R | 5 |
| | U3 | K4 | IO,A18 | 12 |
| | U9 | 112 | OE#_R | 5 |
| LLL49 | U3 | L4 | IO,A20 | 12 |
| | U9 | 108 | IO14_R | 5 |
| LLL50 | U3 | M4 | IO | 12 |
| | U9 | 106 | IO12_R | 5 |
| LLL51 | U3 | N4 | IO | 12 |
| | U9 | 87 | IO2_R | 5 |
| LLL52 | U10 | 93 | IO6_R | 5 |
| | U3 | P4 | IO | 12 |
| LLL53 | U3 | R4 | IO | 12 |
| | U9 | 86 | IO1_R | 5 |
| LLL54 | U10 | 89 | IO3_R | 5 |
| | U3 | T4 | IO | 12 |
| LLL55 | U10 | 10 | A4_R | 5 |
| | U3 | F5 | IO | 12 |
| | U9 | 10 | A4_R | 5 |
| LLL56 | U10 | 11 | A3_R | 5 |
| | U3 | G5 | IO,A12 | 12 |
| | U9 | 11 | A3_R | 5 |
| LLL57 | U10 | 9 | A5_R | 5 |
| | U3 | H5 | IO | 12 |
| | U9 | 9 | A5_R | 5 |
| LLL58 | U10 | 125 | A13_R | 5 |
| | U3 | J5 | IO | 12 |
| | U9 | 125 | A13_R | 5 |
| LLL59 | U10 | 113 | R/W#_R | 5 |
| | U3 | K5 | IO,A19 | 12 |
| | U9 | 113 | R/W#_R | 5 |
| LLL60 | U10 | 108 | IO14_R | 5 |
| | U3 | L5 | IO | 12 |
| LLL61 | U10 | 106 | IO12_R | 5 |
| | U3 | M5 | IO | 12 |
| LLL62 | U10 | 85 | IO0_R | 5 |
| | U3 | N5 | IO | 12 |
| LLL63 | U3 | P5 | IO | 12 |
| | U9 | 93 | IO6_R | 5 |
| LLL64 | U10 | 102 | IO10_R | 5 |
| | U3 | R5 | IO | 12 |
| HHH0 | ST4 | B3 | 3 | 9 |
| | U4 | A3 | IO | 9 |
| HHH1 | ST4 | C4 | 4 | 9 |
| | U4 | A4 | IO | 9 |
| HHH2 | ST4 | B7 | 7 | 9 |
| | U4 | A7 | IO | 9 |
| HHH3 | ST4 | C9 | 9 | 9 |
| | U4 | A8 | IO | 9 |
| HHH4 | ST4 | A10 | 10 | 9 |
| | U4 | A9 | IO | 9 |
| HHH5 | ST4 | B15 | 15 | 9 |
| | U4 | A12 | IO | 9 |
| HHH6 | ST4 | C17 | 17 | 9 |
| | U4 | A13 | IO | 9 |
| HHH7 | ST4 | C19 | 19 | 9 |
| | U4 | A14 | IO | 9 |

| | | | | |
|---|---|---|---|---|
| MMM8 | ST4 | C22 | 22 | 9 |
| | U4 | A17 | IO | 9 |
| MMM9 | ST4 | A22 | 22 | 9 |
| | U4 | A18 | IO | 9 |
| MMM10 | ST4 | C3 | 3 | 9 |
| | U4 | B3 | IO | 9 |
| MMM11 | ST4 | A3 | 3 | 9 |
| | U4 | B4 | IO | 9 |
| MMM12 | ST4 | A4 | 4 | 9 |
| | U4 | B5 | IO | 9 |
| MMM13 | ST4 | C6 | 6 | 9 |
| | U4 | B6 | IO | 9 |
| MMM14 | ST4 | C7 | 7 | 9 |
| | U4 | B7 | IO | 9 |
| MMM15 | ST4 | A8 | 8 | 9 |
| | U4 | B8 | IO | 9 |
| MMM16 | ST4 | B10 | 10 | 9 |
| | U4 | B9 | IO | 9 |
| MMM17 | ST4 | C12 | 12 | 9 |
| | U4 | B10 | IO | 9 |
| MMM18 | ST4 | B13 | 13 | 9 |
| | U4 | B11 | IO | 9 |
| MMM19 | ST4 | C15 | 15 | 9 |
| | U4 | B12 | IO | 9 |
| MMM20 | ST4 | A16 | 16 | 9 |
| | U4 | B13 | IO | 9 |
| MMM21 | ST4 | B18 | 18 | 9 |
| | U4 | B14 | IO | 9 |
| MMM22 | ST4 | B20 | 20 | 9 |
| | U4 | B15 | IO | 9 |
| MMM23 | ST4 | B21 | 21 | 9 |
| | U4 | B16 | IO | 9 |
| MMM24 | ST4 | A21 | 21 | 9 |
| | U4 | B17 | IO | 9 |
| MMM25 | ST4 | B22 | 22 | 9 |
| | U4 | B18 | IO | 9 |
| MMM28 | ST4 | B4 | 4 | 9 |
| | U4 | C5 | IO | 9 |
| MMM29 | ST4 | A5 | 5 | 9 |
| | U4 | C6 | IO | 9 |
| MMM31 | ST4 | B8 | 8 | 9 |
| | U4 | C8 | IO | 9 |
| MMM32 | ST4 | C10 | 10 | 9 |
| | U4 | C9 | IO | 9 |
| MMM33 | ST4 | A11 | 11 | 9 |
| | U4 | C10 | IO | 9 |
| MMM34 | ST4 | C13 | 13 | 9 |
| | U4 | C11 | IO | 9 |
| MMM35 | ST4 | B14 | 14 | 9 |
| | U4 | C12 | IO | 9 |
| MMM36 | ST4 | B16 | 16 | 9 |
| | U4 | C13 | IO | 9 |
| MMM37 | ST4 | C18 | 18 | 9 |
| | U4 | C14 | IO | 9 |
| MMM38 | ST4 | C20 | 20 | 9 |
| | U4 | C15 | IO | 9 |
| MMM39 | ST4 | C21 | 21 | 9 |
| | U4 | C16 | IO | 9 |
| MMM42 | ST4 | B5 | 5 | 9 |
| | U4 | D6 | IO | 9 |
| MMM43 | ST4 | A6 | 6 | 9 |
| | U4 | D7 | IO | 9 |
| MMM44 | ST4 | C8 | 8 | 9 |
| | U4 | D8 | IO | 9 |
| MMM45 | ST4 | A9 | 9 | 9 |
| | U4 | D9 | IO | 9 |
| MMM46 | ST4 | B11 | 11 | 9 |
| | U4 | D10 | IO | 9 |
| MMM47 | ST4 | A12 | 12 | 9 |
| | U4 | D11 | IO | 9 |
| MMM48 | ST4 | C14 | 14 | 9 |
| | U4 | D12 | IO | 9 |
| MMM49 | ST4 | C16 | 16 | 9 |
| | U4 | D13 | IO | 9 |
| MMM50 | ST4 | A17 | 17 | 9 |
| | U4 | D14 | IO | 9 |
| MMM51 | ST4 | A19 | 19 | 9 |
| | U4 | D15 | IO | 9 |
| MMM52 | ST4 | A20 | 20 | 9 |
| | U4 | D16 | IO | 9 |
| MMM53 | ST4 | C5 | 5 | 9 |
| | U4 | E6 | IO | 9 |
| MMM54 | ST4 | B6 | 6 | 9 |
| | U4 | E7 | IO | 9 |
| MMM55 | ST4 | A7 | 7 | 9 |
| | U4 | E8 | IO | 9 |
| MMM56 | ST4 | B9 | 9 | 9 |
| | U4 | E9 | IO | 9 |
| MMM57 | ST4 | C11 | 11 | 9 |
| | U4 | E10 | IO | 9 |
| MMM58 | ST4 | B12 | 12 | 9 |
| | U4 | E11 | IO | 9 |
| MMM59 | ST4 | A13 | 13 | 9 |
| | U4 | E12 | IO | 9 |
| MMM60 | ST4 | A15 | 15 | 9 |
| | U4 | E13 | IO | 9 |
| MMM61 | ST4 | B17 | 17 | 9 |
| | U4 | E14 | IO | 9 |
| MMM62 | ST4 | B19 | 19 | 9 |
| | U4 | E15 | IO | 9 |
| N$1 | R35 | 1 | 1 | 2 |
| | S1 | 1 | 1 | 2 |
| | S1 | 2 | 2 | 2 |
| N$2 | D3 | K | K | 2 |
| | R2 | 2 | 2 | 2 |
| N$3 | IC3 | 5 | Y | 1 |
| | IC6 | 11 | IN_B | 1 |
| | IC7 | 10 | IN_A | 1 |
| | IC7 | 11 | IN_B | 1 |
| N$4 | D4 | K | K | 8 |
| | R39 | 2 | 2 | 8 |
| N$5 | IC4 | 6 | O | 3 |
| | U8 | P22 | CS# | 3 |
| N$7 | J3 | 13 | PROG | 2 |
| | JP11 | 1 | 1 | 2 |
| | JP3 | 2 | 2 | 2 |
| | U3 | U17 | PROGRAM | 2 |
| N$8 | IC4 | 3 | O | 3 |
| | U7 | P22 | CS# | 3 |
| N$9 | J2 | 13 | PROG | 2 |
| | JP3 | 1 | 1 | 2 |
| | JP4 | 2 | 2 | 2 |
| | U2 | U17 | PROGRAM | 2 |
| N$10 | J3 | 9 | DONE | 2 |
| | JP12 | 1 | 1 | 2 |
| | JP5 | 2 | 2 | 2 |
| | U3 | V18 | DONE | 2 |
| N$11 | IC1 | 6 | CE#_OUT | 3 |
| | IC4 | 12 | IO | 3 |
| | IC4 | 13 | I1 | 3 |
| N$12 | IC2 | 10 | B | 1 |
| | JP15 | 2 | 2 | 1 |
| N$13 | J2 | 9 | DONE | 2 |
| | JP10 | 2 | 2 | 2 |
| | JP5 | 1 | 1 | 2 |
| | U2 | V18 | DONE | 2 |
| N$14 | J3 | 2 | RT | 2 |
| | R5 | 2 | 2 | 2 |
| | U3 | C18 | I,M0 | 2 |
| N$15 | IC2 | 11 | A | 1 |
| | JP14 | 2 | 2 | 1 |
| N$16 | IC3 | 9 | C | 1 |
| | JP17 | 2 | 2 | 1 |
| N$17 | J3 | 4 | RD | 2 |
| | R4 | 2 | 2 | 2 |
| | U3 | A20 | O,M1 | 2 |
| N$18 | J1 | 4 | RD | 2 |
| | R33 | 2 | 2 | 2 |
| | U1 | A20 | O,M1 | 2 |
| N$19 | IC3 | 10 | B | 1 |
| | JP16 | 2 | 2 | 1 |
| N$20 | J1 | 2 | RT | 2 |
| | R32 | 2 | 2 | 2 |
| | U1 | C18 | I,M0 | 2 |
| N$21 | JP35 | 2 | 2 | 2 |
| | R32 | 1 | 1 | 2 |
| N$22 | JP36 | 2 | 2 | 2 |
| | R33 | 1 | 1 | 2 |
| N$24 | R34 | 2 | 2 | 2 |
| | U1 | D17 | I,M2 | 2 |
| N$25 | JP37 | 2 | 2 | 2 |
| | R34 | 1 | 1 | 2 |
| N$26 | J2 | 4 | RD | 2 |
| | R1 | 2 | 2 | 2 |
| | U2 | A20 | O,M1 | 2 |
| N$27 | J4 | 13 | PROG | 2 |
| | JP11 | 2 | 2 | 2 |
| | U4 | U17 | PROGRAM | 2 |
| N$28 | J2 | 2 | RT | 2 |
| | R3 | 2 | 2 | 2 |
| | U2 | C18 | I,M0 | 2 |
| N$29 | J4 | 9 | DONE | 2 |
| | JP12 | 2 | 2 | 2 |
| | U4 | V18 | DONE | 2 |
| N$32 | IC3 | 11 | A | 1 |
| | JP30 | 2 | 2 | 1 |
| N$33 | D2 | K | K | 10 |
| | R10 | 2 | 2 | 10 |
| N$44 | IC4 | 4 | IO | 3 |
| | IC4 | 8 | O | 3 |
| N$45 | JP9 | 2 | 2 | 6 |
| | ST1 | A9 | 9 | 6 |
| N$46 | JP18 | 2 | 2 | 6 |
| | ST1 | A11 | 11 | 6 |
| N$48 | JP19 | 1 | 1 | 6 |
| | ST1 | B31 | 31 | 6 |
| N$49 | JP23 | 2 | 2 | 7 |
| | ST2 | B1 | 1 | 7 |
| N$50 | JP24 | 2 | 2 | 7 |
| | ST2 | B12 | 12 | 7 |
| N$51 | JP25 | 2 | 2 | 7 |
| | ST2 | B13 | 13 | 7 |
| N$52 | JP26 | 2 | 2 | 7 |
| | ST2 | B22 | 22 | 7 |
| N$53 | JP20 | 2 | 2 | 6 |
| | ST1 | A17 | 17 | 6 |
| N$55 | JP28 | 2 | 2 | 6 |
| | ST1 | B20 | 20 | 6 |
| N$57 | JP22 | 2 | 2 | 6 |

| | | | | |
|---|---|---|---|---|
| | ST1 | B2 | 2 | 6 |
| N$59 | JP27 | 2 | 2 | 6 |
| | ST1 | B23 | 23 | 6 |
| N$80 | IC4 | 11 | O | 3 |
| | IC4 | 2 | I1 | 3 |
| | IC4 | 5 | I1 | 3 |
| N$110 | IC6 | 2 | OA1 | 1 |
| | R11 | 2 | 2 | 1 |
| N$171 | J4 | 2 | RT | 2 |
| | R7 | 2 | 2 | 2 |
| | U4 | C18 | I,HO | 2 |
| N$172 | J4 | 4 | RD | 2 |
| | R6 | 2 | 2 | 2 |
| | U4 | A20 | O,H1 | 2 |
| N$174 | IC7 | 13 | HON | 1 |
| | R31 | 2 | 2 | 1 |
| N$175 | IC7 | 14 | OB5 | 1 |
| | R30 | 2 | 2 | 1 |
| N$232 | IC7 | 15 | OB4 | 1 |
| | R29 | 2 | 2 | 1 |
| N$233 | IC7 | 17 | OB3 | 1 |
| | R28 | 2 | 2 | 1 |
| N$234 | IC7 | 18 | OB2 | 1 |
| | R27 | 2 | 2 | 1 |
| N$235 | IC7 | 19 | OB1 | 1 |
| | R26 | 2 | 2 | 1 |
| N$236 | IC7 | 7 | OA5 | 1 |
| | R25 | 2 | 2 | 1 |
| N$237 | IC7 | 6 | OA4 | 1 |
| | R24 | 2 | 2 | 1 |
| N$238 | IC7 | 4 | OA3 | 1 |
| | R23 | 2 | 2 | 1 |
| N$239 | IC7 | 3 | OA2 | 1 |
| | R22 | 2 | 2 | 1 |
| N$240 | IC7 | 2 | OA1 | 1 |
| | R21 | 2 | 2 | 1 |
| N$241 | IC6 | 13 | HON | 1 |
| | R20 | 2 | 2 | 1 |
| N$242 | IC6 | 14 | OB5 | 1 |
| | R19 | 2 | 2 | 1 |
| N$243 | IC6 | 15 | OB4 | 1 |
| | R18 | 2 | 2 | 1 |
| N$244 | IC6 | 17 | OB3 | 1 |
| | R17 | 2 | 2 | 1 |
| N$245 | IC6 | 18 | OB2 | 1 |
| | R16 | 2 | 2 | 1 |
| N$246 | IC6 | 19 | OB1 | 1 |
| | R15 | 2 | 2 | 1 |
| N$247 | IC6 | 3 | OA2 | 1 |
| | R12 | 2 | 2 | 1 |
| N$248 | IC6 | 4 | OA3 | 1 |
| | R13 | 2 | 2 | 1 |
| N$249 | IC6 | 6 | OA4 | 1 |
| | R14 | 2 | 2 | 1 |
| NNN2 | ST4 | C28 | 28 | 9 |
| | U4 | G20 | IO | 14 |
| NNN3 | ST4 | A29 | 29 | 9 |
| | U4 | H20 | IO | 14 |
| NNN4 | ST4 | A31 | 31 | 9 |
| | U4 | J20 | IO | 14 |
| NNN12 | ST4 | B23 | 23 | 9 |
| | U4 | D19 | IO | 14 |
| NNN13 | ST4 | C25 | 25 | 9 |
| | U4 | E19 | IO | 14 |
| NNN14 | ST4 | B26 | 26 | 9 |
| | U4 | F19 | IO | 14 |
| NNN15 | ST4 | A27 | 27 | 9 |
| | U4 | G19 | IO | 14 |
| NNN16 | ST4 | B29 | 29 | 9 |
| | U4 | H19 | IO | 14 |
| NNN17 | ST4 | C31 | 31 | 9 |
| | U4 | J19 | IO | 14 |
| NNN28 | ST4 | C23 | 23 | 9 |
| | U4 | D18 | IO | 14 |
| NNN29 | ST4 | A24 | 24 | 9 |
| | U4 | E18 | IO | 14 |
| NNN30 | ST4 | C26 | 26 | 9 |
| | U4 | F18 | IO | 14 |
| NNN31 | ST4 | B27 | 27 | 9 |
| | U4 | G18 | IO | 14 |
| NNN32 | ST4 | C29 | 29 | 9 |
| | U4 | H18 | IO | 14 |
| NNN33 | ST4 | A30 | 30 | 9 |
| | U4 | J18 | IO | 14 |
| NNN34 | ST4 | A32 | 32 | 9 |
| | U4 | K18 | IO | 14 |
| NNN42 | ST4 | B24 | 24 | 9 |
| | U4 | E17 | IO | 14 |
| NNN43 | ST4 | A25 | 25 | 9 |
| | U4 | F17 | IO | 14 |
| NNN44 | ST4 | C27 | 27 | 9 |
| | U4 | G17 | IO | 14 |
| NNN45 | ST4 | A28 | 28 | 9 |
| | U4 | H17 | IO | 14 |
| NNN46 | ST4 | B30 | 30 | 9 |
| | U4 | J17 | IO | 14 |
| NNN47 | ST4 | B32 | 32 | 9 |
| | U4 | K17 | IO | 14 |
| NNN54 | ST4 | B25 | 25 | 9 |
| | U4 | F16 | IO | 14 |
| NNN55 | ST4 | A26 | 26 | 9 |
| | U4 | G16 | IO | 14 |
| NNN56 | ST4 | B28 | 28 | 9 |
| | U4 | H16 | IO | 14 |
| NNN57 | ST4 | C30 | 30 | 9 |
| | U4 | J16 | IO | 14 |
| NNN58 | ST4 | C32 | 32 | 9 |
| | U4 | K16 | IO | 14 |
| PPP2 | U10 | 41 | A12_L | 5 |
| | U4 | P1 | IO | 12 |
| | U9 | 41 | A12_L | 5 |
| PPP3 | U10 | 27 | A2_L | 5 |
| | U4 | N1 | IO,A4 | 12 |
| | U9 | 27 | A2_L | 5 |
| PPP4 | U10 | 51 | CNTRST#_L | 5 |
| | U4 | H1 | IO,A21 | 12 |
| | U9 | 51 | CNTRST#_L | 5 |
| PPP5 | U4 | J1 | IO | 12 |
| | U9 | 58 | IO15_L | 5 |
| PPP6 | U10 | 64 | IO11_L | 5 |
| | U4 | H1 | IO,A10 | 12 |
| PPP7 | U4 | G1 | IO | 12 |
| | U9 | 75 | IO5_L | 5 |
| PPP8 | U4 | D1 | IO | 12 |
| | U9 | 74 | IO6_L | 5 |
| PPP9 | U10 | 74 | IO6_L | 5 |
| | U4 | C1 | IO,A13 | 12 |
| PPP13 | U10 | 42 | A13_L | 5 |
| | U4 | T2 | IO | 12 |
| | U9 | 42 | A13_L | 5 |
| PPP14 | U10 | 43 | A14_L | 5 |
| | U4 | R2 | IO | 12 |
| | U9 | 43 | A14_L | 5 |
| PPP15 | U10 | 23 | CNTEN#_L | 5 |
| | U4 | P2 | IO | 12 |
| | U9 | 23 | CNTEN#_L | 5 |
| PPP16 | U10 | 28 | A3_L | 5 |
| | U4 | N2 | IO | 12 |
| | U9 | 28 | A3_L | 5 |
| PPP17 | U10 | 49 | CE0#_L | 5 |
| | U4 | H2 | IO | 12 |
| | U9 | 49 | CE0#_L | 5 |
| PPP18 | U4 | L2 | IO,A7 | 12 |
| | U9 | 82 | IO0_L | 5 |
| PPP19 | U4 | K2 | IO,A8 | 12 |
| | U9 | 79 | IO2_L | 5 |
| PPP20 | U4 | J2 | IO | 12 |
| | U9 | 76 | IO4_L | 5 |
| PPP21 | U4 | H2 | IO | 12 |
| | U9 | 64 | IO11_L | 5 |
| PPP22 | U10 | 73 | IO7_L | 5 |
| | U4 | G2 | IO | 12 |
| PPP23 | U4 | F2 | IO | 12 |
| | U9 | 61 | IO12_L | 5 |
| PPP24 | U10 | 65 | IO10_L | 5 |
| | U4 | E2 | IO | 12 |
| PPP25 | U10 | 69 | IO8_L | 5 |
| | U4 | D2 | IO | 12 |
| PPP30 | U10 | 21 | ADS#_L | 5 |
| | U4 | R3 | IO | 12 |
| | U9 | 21 | ADS#_L | 5 |
| PPP31 | U10 | 30 | A5_L | 5 |
| | U4 | P3 | IO | 12 |
| | U9 | 30 | A5_L | 5 |
| PPP32 | U10 | 29 | A4_L | 5 |
| | U4 | N3 | IO | 12 |
| | U9 | 29 | A4_L | 5 |
| PPP33 | U10 | 47 | LB#_L | 5 |
| | U10 | 48 | UB#_L | 5 |
| | U4 | H3 | IO,A5 | 12 |
| | U9 | 47 | LB#_L | 5 |
| | U9 | 48 | UB#_L | 5 |
| PPP34 | U10 | 82 | IO0_L | 5 |
| | U4 | L3 | IO,A6 | 12 |
| PPP35 | U10 | 79 | IO2_L | 5 |
| | U4 | K3 | IO,A9 | 12 |
| PPP36 | U10 | 76 | IO4_L | 5 |
| | U4 | J3 | IO,A11 | 12 |
| PPP37 | U10 | 60 | IO13_L | 5 |
| | U4 | H3 | IO | 12 |
| PPP38 | U4 | G3 | IO | 12 |
| | U9 | 73 | IO7_L | 5 |
| PPP39 | U10 | 59 | IO14_L | 5 |
| | U4 | F3 | IO | 12 |
| PPP40 | U4 | E3 | IO | 12 |
| | U9 | 65 | IO10_L | 5 |
| PPP41 | U4 | D3 | IO,A14 | 12 |
| | U9 | 69 | IO8_L | 5 |
| PPP44 | U10 | 32 | A7_L | 5 |
| | U4 | R4 | IO | 12 |
| | U9 | 32 | A7_L | 5 |
| PPP45 | U10 | 33 | A8_L | 5 |
| | U4 | P4 | IO | 12 |
| | U9 | 33 | A8_L | 5 |
| PPP46 | U10 | 31 | A6_L | 5 |
| | U4 | N4 | IO | 12 |
| | U9 | 31 | A6_L | 5 |
| PPP47 | U10 | 25 | A0_L | 5 |
| | U4 | H4 | IO | 12 |

| | | | | |
|---|---|---|---|---|
| | U9 | 25 | A0_L | 5 |
| PPP48 | U10 | 55 | OE#_L | 5 |
| | U4 | L4 | IO,A20 | 12 |
| | U9 | 55 | OE#_L | 5 |
| PPP49 | U4 | K4 | IO,A18 | 12 |
| | U9 | 81 | IO1_L | 5 |
| PPP50 | U4 | J4 | IO | 12 |
| | U9 | 78 | IO3_L | 5 |
| PPP51 | U4 | H4 | IO | 12 |
| | U9 | 60 | IO13_L | 5 |
| PPP52 | U10 | 66 | IO9_L | 5 |
| | U4 | G4 | IO | 12 |
| PPP53 | U4 | F4 | IO | 12 |
| | U9 | 59 | IO14_L | 5 |
| PPP54 | U10 | 61 | IO12_L | 5 |
| | U4 | E4 | IO | 12 |
| PPP55 | U10 | 39 | A10_L | 5 |
| | U4 | R5 | IO | 12 |
| | U9 | 39 | A10_L | 5 |
| PPP56 | U10 | 40 | A11_L | 5 |
| | U4 | P5 | IO | 12 |
| | U9 | 40 | A11_L | 5 |
| PPP57 | U10 | 34 | A9_L | 5 |
| | U4 | N5 | IO | 12 |
| | U9 | 34 | A9_L | 5 |
| PPP58 | U10 | 26 | A1_L | 5 |
| | U4 | M5 | IO | 12 |
| | U9 | 26 | A1_L | 5 |
| PPP59 | U10 | 54 | R/W#_L | 5 |
| | U4 | L5 | IO | 12 |
| | U9 | 54 | R/W#_L | 5 |
| PPP60 | U10 | 81 | IO1_L | 5 |
| | U4 | K5 | IO,A19 | 12 |
| PPP61 | U10 | 78 | IO3_L | 5 |
| | U4 | J5 | IO | 12 |
| PPP62 | U10 | 58 | IO15_L | 5 |
| | U4 | H5 | IO | 12 |
| PPP63 | U4 | G5 | IO,A12 | 12 |
| | U9 | 66 | IO9_L | 5 |
| PPP64 | U10 | 75 | IO5_L | 5 |
| | U4 | F5 | IO | 12 |
| PROGRAM | D1 | K | K | 2 |
| | J1 | 13 | PROG | 2 |
| | R40 | 1 | 1 | 2 |
| | S2 | 1 | 1 | 2 |
| | S2 | 2 | 2 | 2 |
| | U1 | U17 | PROGRAM | 2 |
| QQQ0 | U2 | X3 | IO | 12 |
| | U4 | R16 | IO | 14 |
| QQQ1 | U2 | X4 | IO | 12 |
| | U4 | T17 | IO | 14 |
| QQQ2 | U2 | W3 | IO | 12 |
| | U4 | U18 | IO | 14 |
| QQQ3 | U2 | W4 | IO | 12 |
| | U4 | U20 | IO | 14 |
| QQQ4 | U2 | W5 | IO | 12 |
| | U4 | U19 | IO | 14 |
| QQQ5 | U2 | V5 | IO,RCLK,RB | 12 |
| | U4 | W2 | IO,A0,WS | 10 |
| QQQ6 | U2 | U6 | IO,D1 | 12 |
| | U4 | V7 | IO | 10 |
| QQQ7 | U2 | T6 | IO | 12 |
| | U4 | U5 | IO | 10 |
| RST_CFG | J1 | 17 | RST | 2 |
| | J2 | 17 | RST | 2 |
| | J3 | 17 | RST | 2 |
| | J4 | 17 | RST | 2 |
| | JP13 | 1 | 1 | 2 |
| | R35 | 2 | 2 | 2 |

| | | | | |
|---|---|---|---|---|
| | R42 | 1 | 1 | 2 |
| RST_LCA | JP13 | 2 | 2 | 2 |
| | U1 | A12 | IO | 10 |
| | U2 | X14 | IO,CS0 | 12 |
| | U3 | A14 | IO | 12 |
| | U4 | X12 | IO | 10 |
| SL_CCLK | JP39 | 1 | 1 | 2 |
| | U1 | T19 | IO | 6 |
| SL_PRG | D1 | A | A | 2 |
| | JP4 | 1 | 1 | 2 |
| | R43 | 1 | 1 | 2 |
| | U1 | J20 | IO | 6 |
| TCK | J1 | 12 | TCK | 2 |
| | J2 | 12 | TCK | 2 |
| | U1 | C4 | IO,TCK | 10 |
| | U2 | C4 | IO,TCK | 7 |
| | U3 | C4 | IO,TCK | 12 |
| | U4 | C4 | IO,TCK | 9 |
| TDI1 | J1 | 10 | TDI | 2 |
| | U1 | D5 | IO,TDI | 10 |
| TDI2 | J2 | 10 | TDI | 2 |
| | U1 | U4 | O,TDO | 6 |
| | U2 | D5 | IO,TDI | 7 |
| TDI3 | U2 | U4 | O,TDO | 12 |
| | U3 | D5 | IO,TDI | 12 |
| TDI4 | U3 | U4 | O,TDO | 8 |
| | U4 | D5 | IO,TDI | 9 |
| TDO4 | JP31 | 1 | 1 | 10 |
| | R10 | 1 | 1 | 10 |
| | U4 | U4 | O,TDO | 10 |
| THS | J1 | 14 | THS | 2 |
| | J2 | 14 | THS | 2 |
| | U1 | C7 | IO,THS | 10 |
| | U2 | C7 | IO,THS | 7 |
| | U3 | C7 | IO,THS | 12 |
| | U4 | C7 | IO,THS | 9 |
| VCC | C1 | 2 | 2 | 15 |
| | C10 | 2 | 2 | 15 |
| | C11 | 2 | 2 | 15 |
| | C12 | 2 | 2 | 15 |
| | C13 | 2 | 2 | 15 |
| | C14 | 2 | 2 | 15 |
| | C15 | 2 | 2 | 15 |
| | C16 | 2 | 2 | 15 |
| | C17 | 2 | 2 | 15 |
| | C18 | 2 | 2 | 15 |
| | C19 | 2 | 2 | 15 |
| | C2 | 2 | 2 | 15 |
| | C20 | 2 | 2 | 15 |
| | C21 | 2 | 2 | 15 |
| | C22 | 2 | 2 | 15 |
| | C23 | 2 | 2 | 15 |
| | C24 | 2 | 2 | 15 |
| | C25 | 2 | 2 | 15 |
| | C26 | 2 | 2 | 15 |
| | C27 | 2 | 2 | 15 |
| | C28 | 2 | 2 | 15 |
| | C29 | 2 | 2 | 15 |
| | C3 | 2 | 2 | 15 |
| | C30 | 2 | 2 | 15 |
| | C31 | 2 | 2 | 15 |
| | C32 | 2 | 2 | 15 |
| | C33 | 2 | 2 | 15 |
| | C34 | 2 | 2 | 15 |
| | C35 | 2 | 2 | 15 |
| | C36 | 2 | 2 | 15 |
| | C37 | 2 | 2 | 15 |
| | C38 | 2 | 2 | 15 |
| | C39 | 2 | 2 | 15 |
| | C4 | 2 | 2 | 15 |
| | C40 | 2 | 2 | 15 |
| | C41 | 2 | 2 | 15 |
| | C42 | 2 | 2 | 15 |
| | C43 | 2 | 2 | 15 |
| | C44 | 2 | 2 | 15 |
| | C45 | 2 | 2 | 15 |
| | C46 | 2 | 2 | 15 |
| | C47 | 2 | 2 | 15 |
| | C48 | 2 | 2 | 15 |
| | C49 | 2 | 2 | 15 |
| | C5 | 2 | 2 | 15 |
| | C50 | 2 | 2 | 15 |
| | C51 | 2 | 2 | 15 |
| | C52 | 2 | 2 | 15 |
| | C53 | 2 | 2 | 15 |

| | | | | |
|---|---|---|---|---|
| | C54 | 2 | 2 | 15 |
| | C55 | 2 | 2 | 15 |
| | C56 | 2 | 2 | 15 |
| | C57 | 2 | 2 | 15 |
| | C58 | 2 | 2 | 15 |
| | C59 | 2 | 2 | 15 |
| | C6 | 2 | 2 | 15 |
| | C60 | 2 | 2 | 15 |
| | C61 | 2 | 2 | 15 |
| | C62 | 2 | 2 | 15 |
| | C63 | 2 | 2 | 15 |
| | C64 | 2 | 2 | 15 |
| | C65 | 2 | 2 | 15 |
| | C66 | 2 | 2 | 15 |
| | C67 | 2 | 2 | 15 |
| | C68 | 2 | 2 | 15 |
| | C69 | 2 | 2 | 15 |
| | C7 | 2 | 2 | 15 |
| | C70 | 2 | 2 | 15 |
| | C71 | 2 | 2 | 15 |
| | C72 | 2 | 2 | 15 |
| | C73 | 2 | 2 | 15 |
| | C74 | 2 | 2 | 15 |
| | C75 | 2 | 2 | 15 |
| | C76 | 2 | 2 | 15 |
| | C77 | 2 | 2 | 15 |
| | C78 | 2 | 2 | 15 |
| | C8 | 2 | 2 | 15 |
| | C80 | 2 | 2 | 15 |
| | C81 | 2 | 2 | 15 |
| | C82 | 2 | 2 | 15 |
| | C86 | 2 | 2 | 1 |
| | C87 | 2 | 2 | 1 |
| | C89 | 2 | 2 | 15 |
| | C9 | 2 | 2 | 15 |
| | C90 | 2 | 2 | 15 |
| | C91 | 2 | 2 | 15 |
| | C92 | 2 | 2 | 3 |
| | C96 | 1 | 1 | 6 |
| | D2 | A | A | 10 |
| | D3 | A | A | 2 |
| | D4 | A | A | 8 |
| | IC1 | 2 | VCC | 3 |
| | IC6 | 1 | VCC | 1 |
| | IC6 | 20 | VCC | 1 |
| | IC7 | 1 | VCC | 1 |
| | IC7 | 20 | VCC | 1 |
| | J1 | 1 | VCC | 2 |
| | J2 | 1 | VCC | 2 |
| | J3 | 1 | VCC | 2 |
| | J4 | 1 | VCC | 2 |
| | JP21 | 31 | 31 | 3 |
| | JP32 | 2 | 2 | 6 |
| | JP35 | 3 | 3 | 2 |
| | JP36 | 3 | 3 | 2 |
| | JP37 | 3 | 3 | 2 |
| | R1 | 1 | 1 | 2 |
| | R3 | 1 | 1 | 2 |
| | R4 | 1 | 1 | 2 |
| | R40 | 2 | 2 | 2 |
| | R41 | 2 | 2 | 2 |
| | R42 | 2 | 2 | 2 |
| | R43 | 2 | 2 | 2 |
| | R44 | 2 | 2 | 2 |
| | R5 | 1 | 1 | 2 |
| | R6 | 1 | 1 | 2 |
| | R7 | 1 | 1 | 2 |
| | R9 | 2 | 2 | 4 |
| | U1 | A11 | VCC | 11 |
| | U1 | A16 | VCC | 11 |
| | U1 | A2 | VCC | 11 |
| | U1 | A6 | VCC | 11 |
| | U1 | B20 | VCC | 11 |
| | U1 | E1 | VCC | 11 |
| | U1 | E5 | VCC | 11 |
| | U1 | F20 | VCC | 11 |
| | U1 | K1 | VCC | 11 |
| | U1 | L20 | VCC | 11 |
| | U1 | R1 | VCC | 11 |
| | U1 | T16 | VCC | 11 |
| | U1 | T20 | VCC | 11 |
| | U1 | W1 | VCC | 11 |
| | U1 | X10 | VCC | 11 |
| | U1 | X15 | VCC | 11 |
| | U1 | X19 | VCC | 11 |
| | U1 | X5 | VCC | 11 |
| | U10 | 104 | VCC | 5 |
| | U10 | 105 | VCC | 5 |
| | U10 | 115 | VCC | 5 |
| | U10 | 117 | CE1_R | 5 |
| | U10 | 20 | VCC | 5 |
| | U10 | 50 | CE1_L | 5 |
| | U10 | 52 | VCC | 5 |
| | U10 | 62 | VCC | 5 |
| | U10 | 67 | VCC | 5 |
| | U10 | 72 | VCC | 5 |

| | | | | |
|---|---|---|---|---|
| | U10 | 83 | VCC | 5 |
| | U10 | 88 | VCC | 5 |
| | U10 | 94 | VCC | 5 |
| | U2 | A11 | VCC | 12 |
| | U2 | A16 | VCC | 12 |
| | U2 | A2 | VCC | 12 |
| | U2 | A6 | VCC | 12 |
| | U2 | B20 | VCC | 12 |
| | U2 | E1 | VCC | 12 |
| | U2 | E5 | VCC | 12 |
| | U2 | F20 | VCC | 12 |
| | U2 | K1 | VCC | 12 |
| | U2 | L20 | VCC | 12 |
| | U2 | R1 | VCC | 12 |
| | U2 | T16 | VCC | 12 |
| | U2 | T20 | VCC | 12 |
| | U2 | W1 | VCC | 12 |
| | U2 | X10 | VCC | 12 |
| | U2 | X15 | VCC | 12 |
| | U2 | X19 | VCC | 12 |
| | U2 | X5 | VCC | 12 |
| | U3 | A11 | VCC | 12 |
| | U3 | A16 | VCC | 12 |
| | U3 | A2 | VCC | 12 |
| | U3 | A6 | VCC | 12 |
| | U3 | B20 | VCC | 12 |
| | U3 | E1 | VCC | 12 |
| | U3 | E5 | VCC | 12 |
| | U3 | F20 | VCC | 12 |
| | U3 | K1 | VCC | 12 |
| | U3 | L20 | VCC | 12 |
| | U3 | R1 | VCC | 12 |
| | U3 | T16 | VCC | 12 |
| | U3 | T20 | VCC | 12 |
| | U3 | W1 | VCC | 12 |
| | U3 | X10 | VCC | 12 |
| | U3 | X15 | VCC | 12 |
| | U3 | X19 | VCC | 12 |
| | U3 | X5 | VCC | 12 |
| | U4 | A11 | VCC | 12 |
| | U4 | A16 | VCC | 12 |
| | U4 | A2 | VCC | 12 |
| | U4 | A6 | VCC | 12 |
| | U4 | B20 | VCC | 12 |
| | U4 | E1 | VCC | 12 |
| | U4 | E5 | VCC | 12 |
| | U4 | F20 | VCC | 12 |
| | U4 | K1 | VCC | 12 |
| | U4 | L20 | VCC | 12 |
| | U4 | R1 | VCC | 12 |
| | U4 | T16 | VCC | 12 |
| | U4 | T20 | VCC | 12 |
| | U4 | W1 | VCC | 12 |
| | U4 | X10 | VCC | 12 |
| | U4 | X15 | VCC | 12 |
| | U4 | X19 | VCC | 12 |
| | U4 | X5 | VCC | 12 |
| | U5 | P11 | VDD | 4 |
| | U5 | P15 | VDD | 4 |
| | U5 | P16 | VDD | 4 |
| | U5 | P20 | VDD | 4 |
| | U5 | P27 | VDD | 4 |
| | U5 | P4 | VDD | 4 |
| | U5 | P41 | VDD | 4 |
| | U5 | P54 | VDD | 4 |
| | U5 | P61 | VDD | 4 |
| | U5 | P65 | VDD | 4 |
| | U5 | P70 | VDD | 4 |
| | U5 | P77 | VDD | 4 |
| | U5 | P91 | VDD | 4 |
| | U6 | P11 | VDD | 4 |
| | U6 | P15 | VDD | 4 |
| | U6 | P16 | VDD | 4 |
| | U6 | P20 | VDD | 4 |
| | U6 | P27 | VDD | 4 |
| | U6 | P4 | VDD | 4 |
| | U6 | P41 | VDD | 4 |
| | U6 | P54 | VDD | 4 |
| | U6 | P61 | VDD | 4 |
| | U6 | P65 | VDD | 4 |
| | U6 | P70 | VDD | 4 |
| | U6 | P77 | VDD | 4 |
| | U6 | P91 | VDD | 4 |
| | U6 | P97 | CE2 | 4 |
| | U9 | 104 | VCC | 5 |
| | U9 | 105 | VCC | 5 |
| | U9 | 115 | VCC | 5 |
| | U9 | 117 | CE1_R | 5 |
| | U9 | 20 | VCC | 5 |
| | U9 | 50 | CE1_L | 5 |
| | U9 | 52 | VCC | 5 |
| | U9 | 62 | VCC | 5 |
| | U9 | 67 | VCC | 5 |
| | U9 | 72 | VCC | 5 |
| | U9 | 83 | VCC | 5 |
| | U9 | 88 | VCC | 5 |
| | U9 | 94 | VCC | 5 |
| | X1 | 3 | KL | 6 |
| VCC_REG | IC5 | 2 | OUT | 6 |
| | JP32 | 1 | 1 | 6 |
| WE# | JP21 | 35 | 35 | 3 |
| | R36 | 1 | 1 | 3 |
| | U7 | P29 | WE# | 3 |
| | U8 | P29 | WE# | 3 |

# Anhang D

# Technische Daten

- Logische Kapazität: ca. 110.000 Gatteräquivalente
- On-Board-Speicher:
  - 1024k x 8 Bit asynchrones SRAM für Konfigurationsdaten
  - 256k x 36 Bit Zero Bus Turnaround SRAM, vorbereitet für 1024k x 36 Bit
  - 32k x 32 Bit Dual Port SRAM
- Taktfrequenz: abhängig von der implementierten Schaltung; derzeit begrenzt durch Dual-Port-RAMs (66 MHz)
- Interfaces: 2 x DIN 41612-C96 (VMEbus), 2 x DIN 41612-R96
- Leistungsaufnahme: frequenzabhängig, da CMOS-Logik; max. 20 W
- Betriebsspannung: 5 V oder 3,3 V (intern 3,3 V)
- Abmessungen: Doppeleuropa (160,00 x 233,35 mm), 6lagig

# Literaturverzeichnis


[AC97] Peter Alfke und Bob Conn. *Application Note XAPP 088: I/O Characteristics of the 'XL FPGAs.* Xilinx Inc., San Jose CA, November 1997. URL: `http://www.xilinx.com/xapp/xapp088.pdf`.

[AK96] Adolf Auer und Ralf Kimmelmann. *Schaltungstest mit Boundary Scan.* Hüthig Buch Verlag, Heidelberg, 1996.

[Alf97] Peter Alfke. *Application Note XAPP 090: FPGA Configuration Guidelines.* Xilinx Inc., San Jose CA, November 1997. URL: `http://www.xilinx.com/xapp/xapp090.pdf`.

[BA97] Ray Bittner und Peter Athanas. Wormhole Run-Time Reconfiguration. In *FPGA '97: ACM/SIGDA International Symposium on Field Programmable Gate Arrays.* Special Interest Group on Design Automation (SIGDA), Association for Computing Machinery, 1997.

[BFRV92] Stephen D. Brown, Robert J. Francis, Jonathan Rose und Zvonko G. Vranesic. *Field-Programmable Gate Arrays.* Kluwer Academic Publishers, 1992.

[Buc98] Klaus Buchenrieder. Hardware/Software System Prototyping. In *Design, Automation and Test in Europe (Designer Track)*, Seiten 235–236, Paris, Februar 1998.

[Bur98] Dave Bursky. Next-Generation SRAMs Deliver Higher Performance. *Electronic Design*, 46(15), Juni 1998. URL: `http://www.penton.com/ed/Pages/magSeiten/june2298/digdes/0622de1.htm`.

[CPP+94] J. Champeau, L. Le Pape, B. Pottier, S. Rubini, E. Gautrin und L. Perraudeau. Flexible Parallel FPGA-based Architectures with ArMen. In *Annual Hawaii International Conference on System Sciences (ACM-IEEE)*, Januar 1994.

[Cyp97a] Cypress Semiconductor Corporation, San Jose CA. *CY7C964 Bus Interface Logic Circuit*, Dezember 1997.

[Cyp97b] Cypress Semiconductor Corporation, San Jose CA. *VIC64 VMEbus Interface Controller with D64 Functionality*, Juli 1997.

[Dal98] Dallas Semiconductor, Dallas TX. *DS1250W 3.3V 4096K Nonvolatile SRAM*, preliminary Auflage, 1998.

[DFPR94] P. Dhaussy, J.-M. Filloque, B. Pottier und S. Rubini. ArMen: an FPGA-Based Parallel Architecture. In H. J. Siegel, editor, *International Parallel Processing Symposium (Parallel System Fair)*, Cancùn, Mexico, April 1994.

[Fal98] Heiko Falk. Hardware-Partitionierung für Prototypen-Boards. Diplomarbeit, Universität Dortmund, 1998.

[Fly66] M. J. Flynn. Very High-Speed Computing Systems. In *Proceedings of the IEEE*, Vol. 54, Seiten 1901–1909, Dezember 1966.

[FT93] P. French und R. Taylor. A Self-Reconfiguring Processor. In *IEEE Workshop on FPGAs for custom Computing Machines*, 1993.

[FZI] Fzi-sim-projekt cobra. URL: `http://www.fzi.de/divisions/sim/projects/cobra.html`.

[Gaj94] Daniel D. Gajski. *Specification and Design of Embedded Systems.* Prentice Hall, 1994.

[GDWL92] D. D. Gajski, N. Dutt, A. Wu und S. Lin. *High-Level Synthesis, Introduction to Chip and System Design.* Kluwer Academic Publishers, 1992.

[GN96] P. Graham und B. Nelson. Genetic Algorithms in Software and in Hardware — A Performance Analysis of Workstation and Custom Computing Machine Implementations. In J. Arnold and K. Pocek, editors, *Proceedings of IEEE Workshop on FPGAs for Custom Computing Machines*, Seiten 216–225, Napa, CA, April 1996.

[Guc99] Steve Guccione. List of FPGA-based Computing Machines. URL: `http://www.io.com/~guccione/HW_list.html`, 1999.

[Höf] Wolfgang Höflich. *Using the XC4000 Readback Capability.* Xilinx Inc., San Jose CA. URL: `http://www.xilinx.com/xapp/xapp015.pdf`.

[Hit98] Hitachi Semiconductor Inc., San Jose CA. *HM5264165 / HM5264805 / HM5264405 Series 64M LVTTL Interface SDRAM*, Juli 1998. URL: `http://semiconductor.hitachi.com/products/`.

[IDT] *IDT70V927S/L High-Speed 3.3 V 32K x 16 Synchronous Dual-Port Static RAM.*

[IDT98a] Integrated Device Technology Inc., Santa Clara CA. *Application Note AN-02: Dual-Port SRAMs Simplify Communication in Computer Systems*, 1998.

[IDT98b] Integrated Device Technology Inc., Santa Clara CA. *IDT49FCT3805/A Fast CMOS Buffer / Clock Driver*, Oktober 1998.

[IDT98c] Integrated Device Technology Inc., Santa Clara CA. *IDT70V9279S/L High-Speed 3.3 V 32K x 16 Synchronous Pipelined Dual-Port Static RAM*, preliminiary Auflage, Oktober 1998. URL: `http://www.idt.com/docs/3743.pdf`.

[IDT98d] Integrated Device Technology Inc., Santa Clara CA. *IDT71V547, 128K x 36, CMOS 3.3V Synchronous SRAM with ZBT$^{TM}$ Feature, Burst Counter and Flow-Through Outputs*, preliminary Auflage, April 1998. URL: `http://www.idt.com/docs/3822.pdf`.

[KKR94] G. Koch, U. Kebschull und W. Rosenstiel. A Prototyping Environment for Hardware/Software Codesign in the COBRA Project. In *Proceedings of 3rd international Workshop on Hardware/Software Codesign (CODES/CASHE)*, Grenoble, 1994.

[Lan98] Birger Landwehr. *ILP-basierte Mikroarchitektur-Synthese mit komplexen Bausteinbibliotheken.* Dissertation, Universität Dortmund, 1998.

[Leu97] R. Leupers. Retargetable Codegeneration for Digital Signal Processors. *Dissertation, Universität Dortmund*, 1997.

[Mar93] Peter Marwedel. *Synthese und Simulation von VLSI-Systemen.* Carl Hanser Verlag, 1993.

[Max96] Maxim Integrated Products, Sunnyvale CA. *MAX793 / MAX794 / MAX795 3.0V/3.3V Adjustable Microprocessor Supervisory Circuits*, Januar 1996.

[Mic98] Micron Technology Inc., Boise ID. *Technical Note 05-25: Choosing the Right SRAM*, 1998. URL: `http://www.micron.com`.

[ML96] F. Maier-Lindenberg. The Design of an Efficient Stack Processor on an FPGA Using a New HDL. In *International Workshop on Logic and Architecture Synthesis*, Grenoble, Dezember 1996.

[MLD92] Petra Michel, Ulrich Lauther und Peter Duzy. *The Synthesis Approach to Digital System Design.* Kluwer Academic Publishers, 1992.

[Mot99] Motorola Inc., Schaumburg IL. *FSRAM Product Update: ZBT$^{tm}$ RAMs*, 1999. URL: `http://www.mot.com/SPS/FastSRAM/productupdate/zbt.html`.

[Nie98] Ralf Niemann. *Hardware/Software Codesign for Dataflow-Dominated Embedded Systems.* Dissertation, Universität Dortmund, 1998.

[Par92] Kenneth P. Parker. *The Boundary-Scan Handbook.* Kluwer Academic Publishers, 1992.

[Pet97] Wade D. Peterson. *The VMEbus Handbook.* VMEbus International Trade Association, Scottsdale AZ, 4. Auflage, 1997.

[Pro94] Projektgruppe ENIGMA, Fachbereich Informatik der Universität Dortmund. *Endbericht*, März 1994.

[Pro96] Projektgruppe 260, Fachbereich Informatik der Universität Dortmund. *Endbericht*, März 1996.

[Pro97] Projektgruppe FastFood, Fachbereich Informatik der Universität Dortmund. *Endbericht*, 1997.

[QKSZ94] G. M. Quénot, I. C. Kraljić, J. S'erot und B. Zavidovique. A Reconfigurable Compute Engine for Real-Time Vision Automata Prototyping. In *IEEE Workshop on FPGAs for Custom Computing Machines*, Seiten 91–100, 1994.

[SIA97] Semiconductor Industry Association. National Technology Roadmap for Semiconductors. URL: `http://notes.sematech.org/ntrs/Rdmpmem.nsf`, 1997.

[SJ96] Th. Schwederski und M. Jurczyk. *Verbindungsnetze.* Teubner-Verlag, 1996.

[SW97] J. Staunstrup und W. Wolf, editors. *Hardware/Software Co-Design.* Kluwer Academic Publishers, 1997.

[Syn93] Synopsys Inc. SYNOPSYS User Manuals, 1993.

[TCE+95] E. Tau, D. Chen, I. Eslick, J. Brown und A. DeHon. A First Generation DPGA Implementation. In *Proceedings of the Third Canadian Workshop on Field-Programmable Devices*, Seiten 138–143, Mai 1995.

[Tei97] Jürgen Teich. *Digitale Hardware/Software-Systeme.* Springer-Verlag, 1997.

[Tun98] Tundra Semiconductor Corporation, Canada. *Trooper II$^{TM}$ User Manual*, 1998.

[VME95] VMEbus International Trade Association, Scottsdale AZ. *American National Standard for VME64*, April 1995.

[WH95] M. J. Wirthlin und B. L. Hutchings. A Dynamic Instruction Set Computer. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines.* IEEE Computer Society Press, April 1995.

[Xil94] Xilinx Inc., San Jose CA. *XACT Hardware & Peripherals Guide*, April 1994.

[Xil96] Xilinx Inc., San Jose CA. *The Programmable Logic Data Book*, 1996.

[Xil97] Xilinx Inc., San Jose CA. *Application Note XAPP 017: Boundary Scan in XC4000 and XC5200 Series Devices*, Dezember 1997. URL: `http://www.xilinx.com/xapp/xapp017.pdf`.

[Xil99] Xilinx Inc., San Jose CA. *XC4000E and XC4000X Series Field Programmable Gate Arrays*, Januar 1999. URL: `http://www.xilinx.com/partinfo/4000.pdf`.