Prozessorwahl für das ‘Self Checking RISC Board (SCRIBO)’ *

A. Steininger und J. Reisinger

Institut für Elektrische Meßtechnik
Technische Universität Wien
Gußhausstraße 25
A–1040 Wien, Österreich

Tel.: +43/222/58801–8173, +43/222/58801–3596
Fax: +43/222/5875998
E–mail: joe@vmars.uucp

Version 1.2; 1.6.1990

Zusammenfassung

Dieser Bericht vollzieht die Entscheidungsfindung bei der Prozessorwahl für das ‘Self Checking RISC Board (SCRIBO)’ nach. Aus einer Anzahl von am Markt verfügbaren Prozessoren wird derjenige ausgewählt, der den Anforderungen für den Einsatz in einem fehlertoleranten Echtzeitsystem am besten entspricht.

* Diese Arbeit wird vom Fonds zur Förderung der Wissenschaftlichen Forschung und vom ESPRIT Basic Research Project 3092 ‘Predictably Dependable Computing Systems (PDCS) unterstützt.

Inhaltsverzeichnis

1 Einleitung 2

2 Vorgangsweise bei der Prozessorwahl 2

3 Anforderungen an den Prozessor 2
3.1 Notwendige Eigenschaften 3
3.2 Gewünschte Eigenschaften 3

4 Beschreibung der Prozessoren 4
4.1 Marktübersicht 5
4.2 Nähere Beschreibung der in Frage kommenden Prozessoren 6
4.2.1 Motorola MC88000 6
4.2.2 MIPS R3000 7
4.2.3 Intel i860 7
4.3 Gegenüberstellung 7

5 Entscheidungsfindung 8
5.1 Vor– und Nachteile der Prozessoren 8
5.1.1 Fehlererkennungseigenschaften 8
5.1.2 Leistungsmerkmale 10
5.1.3 Sonstige Merkmale 10
5.2 Auswahl eines Prozessors 11
5.3 Überprüfung der technischen Realisierbarkeit 11

6 Zusammenfassung 11

Literaturreferenzen 13

Anhang A1

Tabellenverzeichnis

Tabelle 1: Gegenüberstellung der Prozessoren 8

1 Einleitung

Am Institut für elektrische Meßtechnik soll im Rahmen eines FWF–Projektes eine experimentelle Untersuchung und Bewertung verschiedener Fehlererkennungsmechanismen in Echtzeitsystemen erfolgen. Als Grundlage für diese Experimente wird ein Prozessorboard entwickelt, das über eine Vielzahl zu– und abschaltbarer Hardwaremechanismen zur Fehlererkennung verfügt. Wegen der zunehmenden Bedeutung von RISC[1] Prozessoren am Markt soll im Rahmen des Projektes diese moderne Architektur eingesetzt werden, um ihre Eignung für fehlertolerante Echtzeitsysteme zu prüfen.

In der ersten Phase des Projekts werden verschiedene RISC–Prozessoren miteinander verglichen, anhand dieses Vergleichs soll der geeignetste Prozessor für das Projekt ermittelt werden. Da von der Prozessorwahl der Erfolg und die Ergebnisse des Projekts maßgeblich beeinflußt werden, kommt dieser Projektphase sehr große Bedeutung zu. Das Hauptaugenmerk des Vergleichs liegt auf den von den Prozessoren unterstützten Fehlererkennungsmechanismen, es müssen aber auch noch Randbedingungen in bezug auf Leistung und Realisierbarkeit (mit vertretbarem Aufwand) erfüllt werden.

2 Vorgangsweise bei der Prozessorwahl

Aus einer Anzahl marktüblicher leistungsfähiger Prozessoren werden zunächst anhand eines Forderungskataloges jene ermittelt, die den gegebenen Rahmenbedingungen überhaupt entsprechen. Anschließend wird unter diesen, durch Abwägen der Vor– und Nachteile aus wissenschaftlicher Sicht ein 'Wunschprozessor' ausgewählt. Für diesen einen Prozessor wird sodann geklärt, ob der technische und finanzielle Aufwand für seinen Einsatz innerhalb vertretbarer Grenzen liegt, ob eine Realisierbarkeit der Wunschlösung mit den vorhandenen Mitteln also gegeben ist. Ist diese nicht gegeben, muß die Entscheidungsfindung neu aufgerollt werden.

3 Anforderungen an den Prozessor

Die Anforderungen werden in zwei Kategorien untergliedert:

Notwendige Eigenschaften sind solche, deren Vorhandensein die Voraussetzung dafür bildet, daß der Prozessor in die engere Wahl gezogen wird. Fehlt auch nur eine dieser Voraussetzungen, so wird der Prozessor aus der weiteren Betrachtung ausgeschlossen.

Gewünschte Eigenschaften sind solche, die vorhanden sein sollen, deren Fehlen jedoch nicht zu einem Ausschluß des Prozessors aus der weiteren Betrachtung führt. Sie dienen einer differenzierten Bewertung jener Prozessoren, die die notwendigen Eigenschaften erfüllen.

1. Reduced Instruction Set Computer

3.1 Notwendige Eigenschaften

- Der Prozessor muß über eine **MMU**[2] verfügen. Dies stellt eine unverzichtbare Grundlage für ein fehlertolerantes System dar (Überwachung des Adreßraums der Tasks!), außerdem ist eine virtuelle Speicherverwaltung höchst wünschenswert für ein Multitaskingsystem, wie es auf den Karten laufen soll.
- Ein 'echter' **RISC**–Prozessor muß verwendet werden. Zielsetzung des Projekts ist auch die Untersuchung der Eignung von RISC–Architekturen für fehlertolerante Echtzeitsysteme, daher werden Prozessoren, die nur einzelne Merkmale einer RISC–Architektur aufweisen, nicht in Betracht gezogen.
- Eine **Fließkommaeinheit** muß entweder auf dem Chip oder als Koprozessor verfügbar sein. Da Echtzeitanwendungen hauptsächlich Steuer– und Regelalgorithmen umfassen, muß eine effiziente Durchführung der dort häufig vorkommenden Fließkommaoperationen gewährleistet sein.
- Die **Leistung** des Prozessors muß mindestens **10 MIPS**[3] betragen. Diese Zahl stellt nur eine ungefähre Richtlinie dar, schon deswegen, weil die'MIPS'–Angaben verschiedener Prozessoren nicht exakt vergleichbar sind. Diese Leistungsangabe dient nur der Vorgabe einer Größenordnung. Da die Karte dem Stand der Technik entsprechen soll, werden Prozessoren mit geringerer Leistung nicht in Betracht gezogen.

  Die Leistung soll eher durch eine **effiziente Prozessorarchitektur** als durch hohe Taktraten erreicht werden (das Verhältnis der Leistung zur Taktfrequenz soll hoch sein). Diese Forderung läßt einerseits Leistungsreserven offen (nachträgliche Erhöhung der Taktrate), andererseits erleichtert sie das Design der Prototypkarte.
- Geeignete **Entwicklungswerkzeuge** müssen allgemein erhältlich sein. Da das Projekt auf eine Kleinserie beschränkt bleibt, ist es nicht möglich, selbst Entwicklungswerkzeuge zu erstellen oder vom Hersteller weitreichende Unterstützung bei der Entwicklung zu erhalten.

3.2 Gewünschte Eigenschaften

Für die Bewertung der Prozessoren werden in einer ersten Ebene ausschließlich wissenschaftliche Kriterien herangezogen. Technische Merkmale, wie z.B. einfaches Design der Schaltung, werden erst in zweiter Ebene betrachtet (bei Prozessoren mit annähernd gleicher wissenschaftlicher Beurteilung).

- Größtmögliche Unterstützung von **Fehlertoleranz**
  - Das **Parity–Bit** soll prozessorintern geprüft werden. Nur durch diese Maßnahme werden auch Fehler erkannt, die bei der Übertragung vom Speicher zum Prozes-

2. Memory Management Unit
3. Million Instructions Per Second

sor — also am Bus — auftreten. Mit einer externen Paritätslogik können im allgemeinen nur vom Speicher fehlerhaft gelieferte Daten erkannt werden; der Bus zwischen Paritätslogik und CPU bleibt ungeschützt.

  - Der Betrieb von zwei Prozessoren in **aktiver Redundanz** soll unterstützt werden. Dieser Mechanismus ermöglicht die Erkennung prozessorinterner Fehler.

Parity–Bit am Prozessor und aktive Redundanz von Prozessoren bilden die Grundlage für ein Design, in dem sämtliche Einfehlerfälle innerhalb der Einheit Speicher–Bus–CPU erkannt werden können. Ist die korrekte Funktion dieser Einheit sichergestellt, so kann auch der korrekte Programmablauf garantiert werden. Dadurch ist es theoretisch möglich, Fehler in allen anderen Funktionseinheiten auf der Karte durch die Software (deren korrekter Ablauf dann ja garantiert ist) zu erkennen. Dadurch sind die Voraussetzungen geschaffen, jeden Einfachfehler auf dem Board zu erkennen.

  - Die Möglichkeit eines **Selbsttests** des Prozessors wäre wünschenswert, da dessen Durchführung vor jedem Systemanlauf die Wahrscheinlichkeit eines nichterkannten permanenten Fehlers im Prozessor verringert.

- **Echtzeitfähigkeit** des Prozessors
  - Die Zeiten für einen **Context Switch** sollen möglichst kurz sein. Dies ist eine generelle Anforderung in Echtzeitsystemen wo ja die Zeitspanne zwischen Stimulus und Response ein wichtiger Parameter ist. In zeitgesteuerten Echtzeitsystemen (MARS [Kop89]) tritt diese Forderung etwas in den Hintergrund. Dort sind die Zeitpunkte der Context–Switches vordefiniert, wodurch die Anzahl der Context–Switches im worst case sehr viel geringer ist als bei ereignisgesteuerten Systemen. Da RISC–Prozessoren generell sehr schnelle Context Switches ermöglichen, ist letztlich nur die Anzahl der zu sichernden Register für die Dauer eines Context Switches ausschlaggebend.
  - Der **garantierte Durchsatz** des Systems soll möglichst hoch sein. In Echtzeitsystemen ist nicht die durchschnittliche, sondern die garantierte Leistung der kritische Parameter, weshalb diesem Punkt besondere Aufmerksamkeit gewidmet werden muß. Die größte Diskrepanz zwischen durchschnittlicher und garantierter Leistung liegt in der Kalkulation der Zugriffe auf den externen Speicher. Bei der Berechnung eines garantierten Busdurchsatzes muß von wesentlich mehr Zugriffen auf den externen Speicher ausgegangen werden als bei der Berechnung eines Durchschnittswertes. Dadurch wird der Busdurchsatz zum wesentlichen Parameter bei der Ermittlung der garantierten Systemleistung.

4 Beschreibung der Prozessoren

In einem ersten Schritt wird Information über die derzeit leistungsfähigsten am Markt erhältlichen Prozessoren gesammelt. Dabei werden sofort die Prozessoren ausgeschieden, die

nicht alle notwendigen Eigenschaften erfüllen. Die restlichen Prozessoren werden detaillierter untersucht.

4.1 Marktübersicht

AMD Am29000: Vom Hersteller als 'Streamlined Instruction Processor' bezeichnet, weist er aber dennoch eine recht RISC–ähnliche Architektur auf. Seine Leistungsdaten sind aber in allen wesentlichen Punkten schlechter als die der 'typischen' RISC–Prozessoren. Da auch die Marktdurchdringung dieses Prozessors äußerst gering ist, wird er nicht näher betrachtet [AMD89].

Cypress CY7C600 (SPARC): Dieser auf der SPARC–Architektur aufbauende Prozessor erfüllt alle notwendigen Anforderungen technischer Art. Es gibt jedoch keine wie immer geartete Unterstützung (Entwicklungswerkzeuge) für Kleinprojekte, daher scheidet er aus [Cyp89].

Inmos IMS T800 (Transputer): Dieser Prozessor wurde für einen eng gekoppelten Betrieb mit gleichartigen Prozessoren entwickelt. Seine Stärken liegen in der effizienten Kommunikation mit anderen Transputern. Er stellt ein eigenes, sehr effizientes Modell zum Context–Switch bereit. Da er eine sehr spezielle Architektur aufweist, die jedenfalls nicht als typische RISC–Architektur bezeichnet werden kann, und er außerdem über keine MMU verfügt, wird er nicht weiter betrachtet [Inm87].

Intel 80486: Einer der gängigsten high–end Prozessoren am Markt. Er besitzt zwar im Gegensatz zu den anderen Prozessoren der 80x86–Serie bereits einige Merkmale der RISC–Architektur, ist jedoch nach wie vor als CISC[4]–Prozessor anzusehen. Aus diesem Grund wird er nicht weiter betrachtet [Cra90].

Intel i860: Der derzeit wohl leistungsfähigste Prozessor auf dem Markt (64 bit Architektur!) erhält alle notwendigen Funktionseinheiten auf einem Chip und erfüllt auch sonst alle Anforderungen [Int89b], [Koh89], [Pie89].

Intel i960CA: In seinem Haupteinsatzgebiet im Bereich der Steuerungen erreicht er durch eine stark parallelisierte Architektur eine sehr hohe Leistung. Er ist jedoch aufgrund der fehlenden MMU für das geplante Board nicht geeignet [Int89], [Int89a].

MIPS R3000: Dieser Prozessor ist als reiner RISC–Prozessor anzusehen. Er erreicht auch bei relativ niedrigen Taktraten (25 MHz) eine hohe Leistung (ca. 21 MIPS). Da er sowohl über eine MMU (intern) als auch über eine Fließkommaeinheit (extern) verfügt, und auch entsprechende Entwicklungswerkzeuge angeboten werden, wird er in die engere Wahl gezogen [Sie89], [Sie89a], [Pie89].

Motorola 68040: Hier gilt sinngemäß dasselbe wie für den 80486 [Ede90].

Motorola MC88000: Er besitzt eine ähnlich leistungsfähige Architektur wie der MIPS R3000. Auch er verfügt über MMU (extern) und Fließkommaeinheit (intern). Entwick-

4. Complex Instruction Set Computer

lungswerkzeuge sind erhältlich. Damit entspricht auch dieser Prozessor allen notwendigen Anforderungen [Mot88], [Mot88a], [Mot88b], [Mot88c], [Pie89].

Nach dieser ersten Betrachtung der Prozessoren bleiben für eine detailliertere Untersuchung noch die Prozessoren *Motorola MC88000*, *MIPS R3000* und *Intel i860* übrig.

4.2 Nähere Beschreibung der in Frage kommenden Prozessoren

Eine Zusammenfassung der Datenblätter der wichtigsten betrachteten Prozessoren ist im Anhang zu finden. Im folgenden werden jene drei Prozessoren, die in die engere Auswahl gekommen sind, hinsichtlich ihrer Fehlererkennungseigenschaften und ihrer Echtzeittauglichkeit untersucht.

4.2.1 Motorola MC88000

- Fehlererkennungsmechanismen
  - Sowohl der zum Datenspeicher als auch der zum Befehlsspeicher führende Bus ist durch Paritätsbits gesichert.
  - Für sämtliche Busleitungen gibt es Komparatoren, die vergleichen, ob die intern angelegten Signalpegel auch extern am Bus erscheinen. Dadurch können z.B. Kurzschlüsse am Bus erkannt werden.
  - Der MC88000 (CPU & MMU) läßt sich im sogenannten 'Checker–Mode' betreiben. Dabei überprüft ein redundant arbeitender Chip (Checker), ob die von ihm erzeugten Ergebnisse mit den Ergebnissen des Master–Chips übereinstimmen. Ist dies nicht der Fall, wird an einem Pin eine Fehlermeldung ausgegeben.
- Der MC88000 unterstützt den Aufbau eines Systems in vollständiger Harvard–Architektur. Das heißt, Daten– und Befehlsspeicher eines Prozessors können physisch getrennt aufgebaut werden. Dieses Architekturmerkmal ermöglicht einen deterministischen Systementwurf in der Hinsicht, daß die Speicherzugriffe für Daten und Befehle nicht miteinander konkurrieren.
- Leistungsmerkmale:
  - ca. 21 MIPS bei 25 MHz
  - Busdurchsatz: 33 MB/s bei Einzelzugriffen (1 Waitstate), 66 MB/s bei Bursts (1,0,0,0 Waitstates) mit 70 ns RAMs. Dieser Durchsatz wird aufgrund der Harvard–Architektur auf jedem der beiden Busse erreicht.
- Der Chip verfügt über 32 32–Bit General Purpose Register.
- Das Cache kann wahlweise im copyback- oder im writethrough-Modus betrieben werden.

- Anzahl der Chips für CPU, FPU, MMU und Cache: 3, Chipfläche: ca. 60cm^2.

4.2.2 MIPS R3000

- Fehlererkennungsmechanismen
  - Der Speicherbus ist durch Paritätsbits gesichert.
- Leistungsmerkmale:
  - ca. 21 MIPS bei 25 MHz
  - Busdurchsatz: 33 MB/s bei Einzelzugriffen (1 Waitstate), 66 MB/s bei Bursts (1,0,0,0 Waitstates) mit 70 ns RAMs. Dieser Durchsatz wird aufgrund eines Zeitmultiplexbetriebs sowohl für Daten als auch für Befehle erreicht, was allerdings auch zu sehr hohen Frequenzen am Bus führt. Es existieren keine getrennten Busse für Zugriffe auf Cache und Hauptspeicher.
- Der Chip verfügt über 32 32–Bit General Purpose Register und 16 64–Bit Fließkommaregister.
- Das Cache wird im writethrough–Modus betrieben.
- Anzahl der Chips für CPU, FPU, MMU und Cache: 12, Chipfläche: ca. 80cm^2.

4.2.3 Intel i860

- Fehlererkennungsmechanismen

  Der i860 verfügt über keine dezidierten Fehlererkennungsmechanismen.
- Leistungsmerkmale:
  - ca. 27 MIPS bei 33 MHz
  - Busdurchsatz: 66 MB/s bei Einzelzugriffen (2 Waitstates), 107 MB/s bei Bursts (2,0,0,0 Waitstates) mit 70 ns RAMs.
    89 MB/s bei Einzelzugriffen (1 Waitstate), 118 MB/s bei Bursts (1,0,0,0 Waitstates) mit 65 ns RAMs.
- Der Chip verfügt über 32 32–Bit General Purpose Register und Fließkommaregister, die wahlweise als 8x128 Bit, 16x64 Bit oder 32x32 Bit organisiert werden können.
- Das Cache wird im copyback–Modus betrieben.
- Anzahl der Chips für CPU, FPU, MMU und Cache: 1, Chipfläche: ca. 20cm^2.

4.3 Gegenüberstellung

In Tabelle 1 werden die wichtigsten Eigenschaften der betrachteten Prozessoren noch einmal zusammengefaßt, um einen unmittelbaren Vergleich zu ermöglichen.

	MC88000	R3000	i860
Parity am Prozessor	ja	ja	nein
Checker Modus	ja	nein	nein
Selbsttest	nein	nein	nein
Leistung	21MIPS/25MHz	21MIPS/25MHz	27MIPS/33MHz
Busdurchsatz/Single	2x33MB/s	2x33MB/s	66MB/s
Busdurchsatz/Burst	2x66MB/s	2x66MB/s	107MB/s
GP–Register	32x32Bit	32x32Bit	32x32Bit
FP–Register	–	16x64Bit	konfigurierbar
Cache Strategie	copyback oder writethrough	writethrough	copyback

Tabelle 1: Gegenüberstellung der Prozessoren

5 Entscheidungsfindung

Im folgenden Kapitel werden, aus rein wissenschaftlicher Sicht, die Fähigkeiten der Prozessoren für den Aufbau eines fehlertoleranten und echtzeitfähigen Systems verglichen. Anhand der Ergebnisse dieser Betrachtung wird anschließend ein Prozessor für das zu konstruierende Board ausgewählt. Eine technische Prüfung erfolgt nur insoweit, als mit dem gewählten Prozessor der Aufbau des Boards innerhalb des durch das Projekt gegebenen Rahmens prinzipiell möglich sein muß. Ein dezidierter Vergleich dieser Eigenschaften als Beitrag zur Entscheidungsfindung ist vom wissenschaftlichen Standpunkt aus nicht zielführend.

5.1 Vor– und Nachteile der Prozessoren

5.1.1 Fehlererkennungseigenschaften

Eines der wichtigsten Ziele des Projekts ist der Aufbau einer Karte mit einem möglichst hohen Grad an Self–Checking Coverage. Daher wäre es wünschenswert, *alle* Funktionseinheiten (Chips, Busse, …) auf der Karte perfekt durch Fehlererkennungsmechanismen abzusichern. Eine konsequente Absicherung gegen Fehler ist aber speziell bei Peripheriebausteinen nicht möglich:

- Passende Peripheriebausteine, die in Redundanz betrieben werden können, sind am Markt nicht verfügbar.

- Marktübliche Bausteine, die Interaktionen mit der Umwelt durchführen, können prinzipiell nicht in aktiver Redundanz betrieben werden. Da die Daten der Peripherie asynchron eintreffen, kann selbst bei noch so genauer Synchronisation der redundanten Peripheriebausteine nicht garantiert werden, daß die Daten innerhalb desselben Taktes übernommen werden; es ist immer möglich, daß Daten zu einem Zeitpunkt eintreffen, wo sie der eine Baustein gerade noch während des laufenden Taktes übernimmt, der andere jedoch erst im nächsten Takt, wodurch bei korrekter Funktion beider Chips dennoch unterschiedliche Ausgaben erzeugt werden können.

Eine direkte Überwachung der Peripheriebausteine ist also nicht möglich, folglich muß eine andere Strategie für die Fehlererkennung vorgesehen werden. Fehler an diesen Bausteinen müssen indirekt anhand der von ihnen gelieferten Ergebnisse erkannt werden. Die Einhaltung verschiedener Zeitbedingungen kann von zusätzlicher Hardware erfolgen werden, eine inhaltliche Kontrolle muß per Software erfolgen.

Die korrekte Funktion der Einheit CPU–Bus–Speicher kann, im Gegensatz dazu direkt durch Hardwaremaßnahmen überwacht werden. Durch Parity–Bits (bis zum Prozessor) und eine redundante CPU kann jeder Einfehlerfall in dieser Einheit lückenlos erkannt werden:

- Der Speicher wird durch das Parity–Bit geschützt.
- Der Prozessor wird durch die redundante Einheit geschützt.
- Der Bus zwischen Prozessor und Speicher kann durch das Parity–Bit nur dann durchgehend geschützt werden, wenn die Parity erst im Prozessor selbst geprüft bzw. generiert wird.

Die korrekte Funktion der Einheit CPU–Bus–Speicher bildet die notwendige Voraussetzung für die zuverlässige Überprüfung der Peripheriebausteine per Software. Nur wenn sie gewährleistet ist, kann auch die korrekte Funktion der überprüfenden Software garantiert werden.

Aus diesen Überlegungen folgt der Wunsch, daß sowohl die Paritätslogik im Prozessor implementiert ist, als auch daß der Prozessor für einen Betrieb in Redundanz geeignet ist. Diese Forderung erfüllt nur der MC88000, während der R3000 lediglich die Paritätsprüfung im Prozessor unterstützt.

Ein weiterer Vorzug des MC88000 gegenüber den anderen beiden Prozessoren ist seine Fähigkeit, mittels interner Komparatoren Kurzschlüsse am Bus zu erkennen. Die Möglichkeit des flexiblen Einsatzes bzw. der wahlweisen Aktivierung der beschriebenen Mechanismen bietet beim MC88000 ein weites Feld für Experimente.

Ein automatischer Selbsttest der Prozessoren wäre wünschenswert, wird jedoch von keinem der drei betrachteten Prozessoren angeboten. Stattdessen muß beim Start der Karten ein Selbsttest durch geeignete Testprogramme durchgeführt werden. Ein automatischer Selbsttest würde zwar bessere Aussagen liefern, er kann aber ebensowenig wie der manuelle Selbsttest vollständig sein.

Vom Standpunkt der Fehlertoleranz aus ist daher der MC88000 den beiden anderen Prozessoren überlegen.

5.1.2 Leistungsmerkmale

Die Leistung der drei betrachteten Prozessoren ist ungefähr gleich. Der i860 bietet eine etwas höhere Leistung, bei einer allerdings auch höheren Taktfrequenz, die jedoch durch die Busarchitektur dieses Prozessors weder Auswirkungen auf die Komplexität des Board–Designs noch auf die Anforderungen an die Speicherbausteine hat. Die zusätzliche Leistung des i860 durch seine interne Grafikeinheit bringt für die geplanten Anwendungen keinen Vorteil.

Die *garantierte* Leistung hängt gerade bei RISC–Prozessoren zu einem Großteil vom Durchsatz am Speicherbus ab. Der i860 trägt dieser Forderung durch einen 64 Bit breiten Bus Rechnung, der MC88000 durch seine Harvard–Architektur und der R3000 durch einen Multiplexbetrieb für Daten und Instruktionen. Alle Prozessoren erreichen auf verschiedenem Weg annähernd den gleichen Busdurchsatz.

Durch die Trennung von Daten– und Befehlsbus beim MC88000 wird die Berechnung des garantierten Busdurchsatzes vereinfacht, da keine Konkurrenz der Zugriffe auf Daten und Befehle zu berücksichtigen ist. Auch beim R3000 sind Zugriffe auf Daten und Befehle unabhängig voneinander durchführbar, jedoch bedeutet der Zeitmultiplexbetrieb eine Verdopplung der am Bus auftretenden Frequenzen und erfordert zudem ein eher unübersichtliches Design.

Da R3000 und i860 neben den General Purpose Registern auch noch über Floating Point Register verfügen, können die Programme etwas effizienter gestaltet werden; allerdings werden dadurch die Zeiten für den Context–Switch geringfügig erhöht.

Bei der durchschnittlichen Leistung gibt es leichte Vorteile für den i860; der Busdurchsatz ist bei allen drei Prozessoren etwa gleich, wobei der Zeitmultiplexbetrieb des R3000 höhere Störanfälligkeit befürchten läßt.

5.1.3 Sonstige Merkmale

- Das Cache beim R3000 wird im writethrough Modus betrieben, beim i860 im copyback Modus. Der MC88000 beherrscht beide Modi. Beim copyback Modus treten zwar im Durchschnitt weniger Buszugriffe auf, der writethrough Modus zeigt aber ein deterministischeres Verhalten. Während bei letzterem jeder Schreibzugriff eins zu eins auf einen Buszugriff abgebildet wird, kann beim copyback Modus ein Schreibzugriff das Leeren einer ganzen Cache–Line bewirken, was zu vier Buszugriffen führt. Daher ist der writethrough Modus im Sinne eines deterministischen Zeitverhaltens besser geeignet.

  Der beim i860 nicht vorhandene writethrough Modus bewirkt ein etwas weniger deterministisches Verhalten dieses Prozessors.

- Da sowohl Chipfläche als auch Anzahl der Chips entscheidenden Einfluß auf die Fehleranfälligkeit der Schaltung haben, soll beides möglichst gering gehalten werden. Diese

Forderung erfüllt der i860 am besten: Sämtliche Funktionen der Zentraleinheit befinden sich auf einem Chip mit einer Fläche von ca. 20 cm^2. Der MC88000 benötigt dafür drei Chips und eine Fläche von ca. 60 cm^2, der R3000 zwölf Chips auf ca. 80 cm^2.

Die geringe Chipfläche des i860 und seine kompakte Bauweise dürfte eine geringere Fehlerrate dieses Prozessors bedingen.

5.2 Auswahl eines Prozessors

Da der R3000 in keinem der angeführten Punkte Vorteile gegenüber den anderen beiden Typen aufweisen kann, muß die Entscheidung zwischen dem i860 und dem MC88000 fallen.

Der Vorteil des i860 ist die geringere Fehleranfälligkeit durch seine kompakte Bauweise. Demgegenüber ist der MC88000 bei den Fehlererkennungsmechanismen und dem deterministischen Zeitverhalten (Harvard Architektur, writethrough Modus) überlegen. Die Möglichkeit des MC88000, *jeden* Einfehlerfall in der Einheit CPU–Bus–Speicher zu erkennen, ist deutlich höher zu bewerten als die beim i860 (möglicherweise) geringere Wahrscheinlichkeit für das Auftreten eines Fehlers (der dann aber *nicht unbedingt* erkannt wird). Da es *nicht* Ziel des Projektes ist, eine möglichst zuverlässige Karte zu produzieren, sondern eine Karte mit möglichst guten Self–Checking Eigenschaften, wird dem **MC88000** der Vorzug gegeben. Das Problem der Zuverlässigkeit kann, aufbauend auf gut funktionierenden Self–Checking Komponenten, durch Redundanz auf Systemebene gelöst werden.

5.3 Überprüfung der technischen Realisierbarkeit

Vor einer endgültigen Entscheidung für den MC88000 wurde noch geprüft, ob sich ein Schaltungsaufbau mit diesem Prozessor im Rahmen der durch das Projekt bereitgestellten Möglichkeiten auch realisieren läßt. Folgende Bedingungen wurden überprüft:

- Entwicklungssoftware (Compiler, Assembler) für vorhandene Systeme muß erhältlich sein.
- Lieferzeiten für Prozessor und passende Peripheriebausteine müssen vertretbar sein.
- Alle Funktionseinheiten müssen auf einem Board sinnvoller Größe unterzubringen sein.

6 Zusammenfassung

Nach eingehender Betrachtung der am Markt erhältlichen RISC–Prozessoren hat sich der Motorola MC88000 eindeutig als der für das geplante Projekt am besten geeignete Prozessor erwiesen. Diese Entscheidung beruht vor allem auf seinen hervorragenden Eigenschaften zur

Fehlererkennung. Nachdem auch die Realisierbarkeit geklärt wurde, wird der MC88000 als Prozessor für das Self–Checking RISC Board gewählt.

Literaturreferenzen

[AMD89] *29K Family Data Book*. Advanced Micro Devices, Sunnyvale, California, 1989.

[Cra90] John. A. Crawford. The i486 CPU: Executing Instructions in One Clock Cycle. *IEEE Micro*, 10(1):27–36, Feb. 1990.

[Cyp89] *Data Book: CMOS, BiCMOS*, Section 6, pages 6–1…6–23. Cypress Semiconductor Corporation, San Jose, CA, Feb. 1989.

[Ede90] Robin W. Edenfield, Michael G. Gallup, William B. Ledbetter, Jr., Ralph C. McGarity, Eric E. Quintana, and Russell A. Reininger. The 68040 Processor, Part 1, Design and Implementation. *IEEE Micro*, 10(1):66–78, Feb. 1990.

[Inm87] *Data Sheet: IMS T800 transputer*. INMOS Limited, Bristol, England, Apr. 1987.

[Int89] *Data Sheet: 80960CA–33, –25, –16 32–Bit High Performance Embedded Processor*. Intel Corporation, Santa Clara, CA, Nov. 1989.

[Int89a] *80960CA User's Manual*. Intel Corporation, Santa Clara, CA, 1989.

[Int89b] *Data Sheet: i860 64–Bit Microprocessor*. Intel Corporation, Santa Clara, CA, Apr. 1989.

[Koh89] Les Kohn and Neal Margulis. Introducing the Intel i860 64–Bit Microprocessor. *IEEE Micro*, 9(4):15–30, Aug. 1989.

[Kop89] H. Kopetz, A. Damm, Ch. Koza, M. Mulazzani, W. Schwabl, Ch. Senft, and R. Zainlinger. Distributed Fault–Tolerant Real–Time Systems: The MARS Approach. *IEEE Micro*, 9(1):25–40, Feb. 1989.

[Mot88] *Technical Summary: 32–Bit RISC Microprozessor MC88100*. Motorola Inc., Phoenix, Arizona, Dec. 1988.

[Mot88a] *Technical Summary: 16–Kilobyte Cache/Memory Management Unit (CMMU) MC88200*. Motorola Inc., Phoenix, Arizona, Dec. 1988.

[Mot88b] *MC88100 User's Manual*. Motorola Inc., Phoenix, Arizona, Nov. 1988.

[Mot88c] *MC88200 User's Manual*. Motorola Inc., Phoenix, Arizona, Nov. 1988.

[Pie89] Richard S. Piepho and William S. Wu. A Comparison of RISC Architectures. *IEEE Micro*, 9(4):51–62, Aug. 1989.

[Sie89] *Data Sheet: SAB R3000 High performance 32–bit RISC Microprocessor*. Siemens AG, München, Germany, Sep. 1989.

[Sie89a] *Data Sheet: SAB R3010 High performance Floating–Point Coprocessor*. Siemens AG, München, Germany, Oct. 1989.