Protokoll vom: 26.1.1995

Seite 1

Projektgruppe Systementwurf 94-95

Lars Werner

Beginn: der Besprechung: 13.00 Uhr
Ende der Besprechung: 14.24 Uhr

Zu Beginn der Besprechung schlägt Ralf Jungblut vor, daß es zweckmäsiger wäre den Protokollführer aus einer Gruppe zu wählen, die nicht vorträgt, da es so nicht allen Teilnehmern der entsprechenden Arbeitsgruppe möglich ist vorzutragen.
Christof Nagel begründet die bisherige Vorgehensweise damit, daß es dem Protokollschreiber erheblich leichter fällt über ein Thema zu protokollieren das ihm bekannt ist, als über ein fremdes Thema.
Es bleibt bei dem bisherigen Bestimmungsverfahren.

Ralf schlägt vor von jedem Projektgruppenteilnehmer 5,- DM einzusammeln, damit Andreas Hennig demnächst zu den Projektgruppenbesprechungen Süßigkeiten organisiert.
Andreas erklärt sich dazu bereit und der Vorschlag wird einstimmig angenommen. Von den 16 Teilnehmern werden zusammen 80,- DM eingesammelt.

Christof teilt mit, daß er bis Dienstag, den 31.1.95 die A30-Papiere der einzelnen Gruppen erhalten möchte.
Die Vorschriften über den genauen Inhalt werden von ihm 3 mal kopiert und in dem Protokollortner abgeheftet.

Die Teilgruppe "Interface" berichtet von ihrer Arbeit der vergangenen Woche.
Es folgen die von dieser Gruppe erstellten Unterlagen:

Nachfolgend findet man Tabellen die die Leitungen aus Abbildung 1 beschreiben:

Signal	Beschreibung SFU Bus
RD[31:0]	Bidirektionale Datenleitung des SFU Busses.
RWA[4:0]	Auswahl des zu beschreibenden Registers.
RRA[1:0]	Auswahl des zu lesenden Ergebnisregisters.
RWR	Wenn diese Leitung aktiv ist (RWR='1'), wird das Datum auf RD in das durch RWA adressierte Register geschrieben und das Valid Bit gesetzt.
RRD	Wenn diese Leitung aktiv ist (RRD='1'), wird der Wert des vorher mit RWA adressierte Register auf den Bus RD gelegt, ansonsten ist der Registerausgang hochohmig.
RVALID	Rechenwerk teilt dem SFU Controller mit, das alle 4 Ergebnisse in den Ausgaberegistern stehen.
RRV	Reset RVALID, setzt das Valid Bit für die Ausgaberegister zurück.

Signal	Beschreibung des Busses zum Rechenkern
VALID	Auf dieser Leitung liegt das Validbit des Registers an.
WRITE[31:0]	Datenleitung vom Rechenkern zum Register, zum schreiben.
READ[31:0]	Datenleitung vom Rechenkern zum Register, zum lesen.
SET	Rechenkern teilt dem Register mit, das es die Daten von WRITE übernehmen soll.
RV	Reset Valid, dieses Signal wird vom Rechenkern benutzt, um das Valid Flag im Register zu löschen.
RESULT[31:0]	Datenleitung vom Rechenkern zum Ausgaberegister
SR	Set Result, damit teilt das Rechenwerk dem Ausgaberegister mit, das es den anliegenden Wert übernehmen soll.
SRV	Set RVALID, setzt das Valid Bit für die Ausgaberegister.

Das SFU-Bus-Protokoll

Zuerst wird er Begiff *aktivieren* erklärt, da er im Folgenden häufiger auftritt. Aktivieren sie demnach wie folgt definiert:
Die entsprechende Leitung soll von einer fallenden Taktflanke, bis zur nächsten fallenden Taktflanke auf logisch '1' gehalten werden und anschließend auf logisch '0' gezogen werden. Die entsprechenden Daten- oder Adressleitungen müssen selbstverständlich ihren Zustand behalten, bis die aktivierte Leitung wieder auf logisch '0' liegt.

Reset:

- Interface:

  Die RRV-Leitung muß für die Dauer eines Systemzyklus auf logisch '1' gelegt werden.

- Rechenkern:

  Die RV-Leitung muß für die Dauer eines Systemzyklus auf logisch '1' gelegt werden.

Input-Register vom Interface beschreiben:

1. Adresse des Registers auf RWA legen
2. Daten auf RD legen
3. zur Datenübernahme RWR aktiviern

Input-Register vom Rechenkern lesen:

1. warten bis VALID auf logisch '1' liegt
2. Daten von den READ-Leitungen des entsprechenden Registern lesen

Input-Register vom Rechenkern schreiben:

1. warten bis VALID auf logisch '1' liegt
2. Daten auf WRITE-Leitungen legen
3. SET aktivieren

Output-Register vom Rechenkern beschreiben:

1. warten bis RVALID auf logisch '0' liegt
2. Daten auf die RESULT-Leitungen legen
3. SET-Result aktivieren
4. wenn das Ergebnis vollständig im Output-Registersatz ist, dann SRV aktivieren

Output-Register vom Interface lesen:

1. warten, bis RVALID auf logisch '1' liegt
2. Registeradresse auf RRA legen
3. RRD aktivieren
4. Daten von RD lesen
5. wenn alle Daten gelesen, dann RRV aktivieren

Ablauf einer Spline Berechnung:

1. Interface beschreibt die Input-Register und signalisiert über das auf '1' gesetzte VALID-Bit dem Rechenkern, daß das jeweilige Register beschrieben wurde.
2. Der Rechenkern erkennt die gesetzten VALID-Bits der Register und beginnt mit der Berechnung. Das Beschreiben der Input-Register vom Rechenkern ist bis zum Rücksetzen des VALID-Bits möglich.
3. Es liegen Ergebnisse vor, die in die Output-Register geschrieben werden können. Diese werden jetzt zyklisch in die Output-Register geschoben. Spätestens wenn alle Ergebnisse geschrieben sind, muß er (Rechenkern) alle VALID-Bits der Input-Register mit dem Aktivieren der entsprechenden RV-Leitung zurücksetzen, damit eine weitere Berechnung möglich wird.
4. Die Ergebnisse in den Output-Registern werden vom Interface gelesen.

Das oben vorgestellete Buskonzept ist selbsterklärend und braucht an dieser Stelle nicht näher erläutert zu werden. Eventuell siehe A30-Papier dieser Gruppe.
Bei dem Vortrag wurde festgestellt, daß es eventuell Timing-Probleme bei der Umschaltung des Multiplexers gibt. Diese beeinflussen das Busprotokoll jedoch nicht, müssen jedoch noch genauer überarbeitet werden.

Die Gruppe "Interface" bemerkt zu der von ihnen entwickelten Schnittstelle daß diese so nicht realisierbar ist, da sie nach einer Hochrechnung ca. 23 000 Gatter des "INKA" Emulators belegt und so für den Rechenkern nur noch ca. 7 000 Gatter vebleiben, da wegen der vielen Leitungen eine Nutzung des 2. Boards nicht möglich ist.
Es wird vorgeschlagen einen von Wolfram Hardt entwickelten RAM-Baustein an das INKA anzuschließen und die Register dort abzulegen.
Dieser Vorschlag wird abgelehnt, da durch den sequentiellen Zugriff auf das RAM die Parallelität der SFU verlorengeht und dadurch die SFU langsamer wird.
Nach längerer Besprechung wird festgestellt, daß eine Reduzierung der Gatter durch Verringerung der implementierten Register und deren Bit-Breite realisiert werden kann. Dies kann jedoch die "Interface"-Gruppe nicht allein entscheiden, da sie die Implementierung des Algorithmus nicht kennt.
Die "Algorithmus"-Gruppe verspricht sich mit diesem Thema in der nächsten Woche auseinander zu setzen und eine Lösung mit geringerem Aufwand zu finden.

Es wird darüber nachgedacht, ob eine Realisierung von RAM im INKA auf eine andere Weise als über Flipflops möglich und sinnvoll ist. Christof will sich darum kümmern.

Edwin stellt eine Möglichkeit vor, wie man eventuell den Algorithmus vereinfachen könnte. Er wird beauftragt seine Idee ausführlich auszuarbeiten und am Donnerstag, den 2.2.95 vorzutragen.

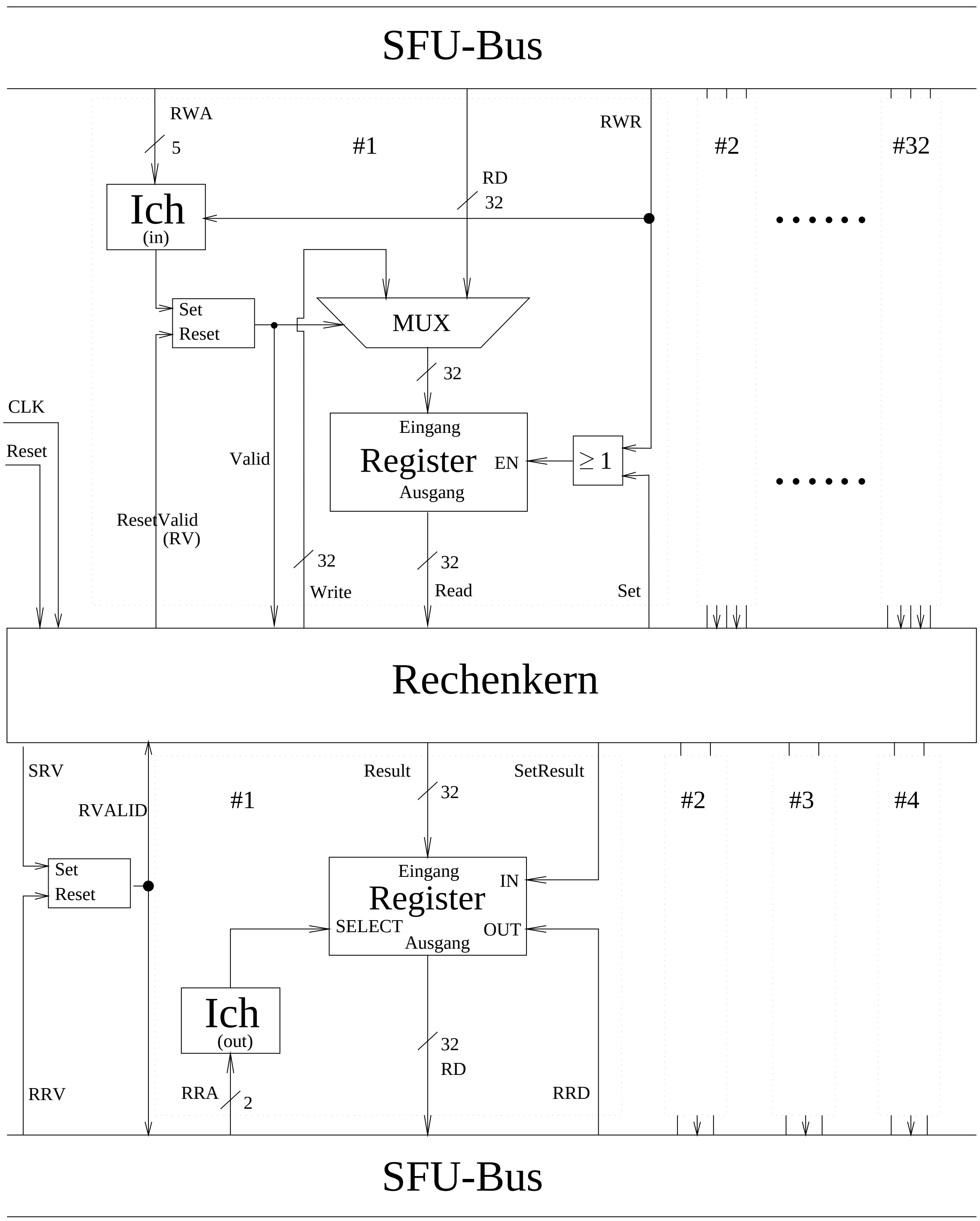

Abbildung 1: Schnittstelle zwischen dem SFU-Bus und zu dem Rechenkern