

Microcode simulation

part 1

Inhoudsopgave

1 Uitleg.....	3
2 Overzicht van de geïmplementeerde instructies.....	4
3 Gegevens pad schema.....	5
4 Gegevens lijn Specificatie.....	6
4.1 Gegevens lijn 1.....	6
4.2 Gegevens lijn 2.....	6
4.3 Gegevens lijn 3.....	7
4.4 Gegevens lijn 4.....	7
4.5 Gegevens lijn 5, 6, 7, 8, 9, 10.....	7
4.6 Gegevens lijn 11, 16.....	8
4.7 Gegevens lijn 12.....	8
4.8 Gegevens lijn 13.....	8
4.9 Gegevens lijn 11, 16.....	9
4.10 Gegevens lijn 15.....	9
4.11 Gegevens lijn 17.....	9
5 Microcode.....	10
5.1 Gedeelte 1.....	10
5.2 Gedeelte 2.....	11

1 Uitleg

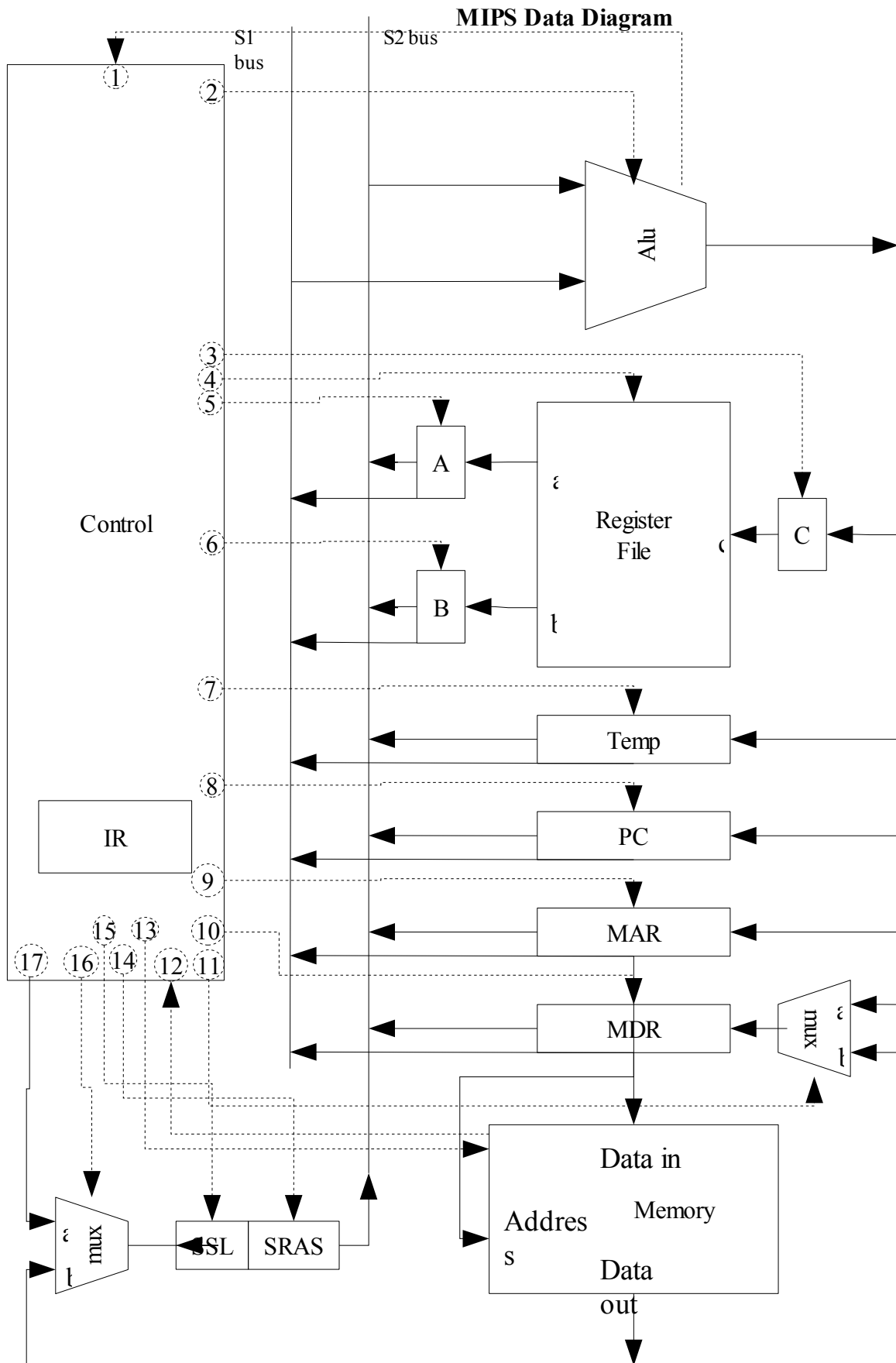
We hebben gekozen om 2 extra blokken te introduceren om de basis invoer zo te kunnen manipuleren zodat hij gelijk is aan de invoer zoals hij uit een register zal komen. Hiermee kunnen wij het aantal codes in de instructieset reduceren, maar moeten wij wel genoegen nemen met het feit dat de processor op een iets langzamer klok cyclus zal komen te draaien

Ons geheugen hebben we een zekere intelligentie toegekend waardoor deze een goed besef heeft hoeveel hij moet lezen en waar dit te plaatsen op de data lijn.

2 Overzicht van de geïmplementeerde instructies

Instruction	opcode		funcode		Type Def.	Type
	hex	bin	hex	bin		
BEQ	4	000100		000000	BRANCH	4DE BIT I-Type
BNE	5	000101		000000	BRANCH	I-Type
ADDI	8	001000		000000	IMMEDIATE	3DE BIT I-Type
ADDIU	9	001001		000000	IMMEDIATE	I-Type
SLTI	0A	001010		000000	IMMEDIATE	I-Type
SLTIU	0B	001011		000000	IMMEDIATE	I-Type
ANDI	0C	001100		000000	IMMEDIATE	I-Type
ORI	0D	001101		000000	IMMEDIATE	I-Type
XORI	0E	001110		000000	IMMEDIATE	I-Type
LUI	0F	001111		000000	IMMEDIATE	I-Type
LH	21	100001		000000	MEM	1STE BIT I-Type
LB	20	100000		000000	MEM	I-Type
LW	23	100011		000000	MEM	I-Type
LBU	24	100100		000000	MEM	I-Type
LHU	25	100101		000000	MEM	I-Type
SB	28	101000		000000	MEM	I-Type
SH	29	101001		000000	MEM	I-Type
SW	2B	101011		000000	MEM	I-Type
SLL		000000	0	000000	REGISTER	NO BIT R-Type
SRL		000000	2	000010	REGISTER	R-Type
SRA		000000	3	000011	REGISTER	R-Type
SLLV		000000	4	000100	REGISTER	R-Type
SRLV		000000	6	000110	REGISTER	R-Type
SRAV		000000	7	000111	REGISTER	R-Type
JR		000000	8	001000	REGISTER	R-Type
JALR		000000	9	001001	REGISTER	R-Type
BREAK		000000	0D	001101	REGISTER	R-Type
ADD		000000	20	100000	REGISTER	R-Type
ADDU		000000	21	100001	REGISTER	R-Type
SUB		000000	22	100010	REGISTER	R-Type
SUBU		000000	23	100011	REGISTER	R-Type
AND		000000	24	100100	REGISTER	R-Type
OR		000000	25	100101	REGISTER	R-Type
XOR		000000	26	100110	REGISTER	R-Type
NOR		000000	27	100111	REGISTER	R-Type
SLT		000000	2A	101010	REGISTER	R-Type
SLTU		000000	2B	101011	REGISTER	R-Type
J	2	000010		000000	JUMP	5DE BIT J-Type
JAL	3	000011		000000	JUMP	J-Type

3 Gegevens pad schema



4 Gegevens lijn Specificatie

4.1 Gegevens lijn 1

<i>Van:</i>	ALU
<i>Naar:</i>	Control
<i>Bit lijnen:</i>	1
<i>Betekenis:</i>	Overflow van de ALU
<i>0</i>	Geen Overflow
<i>1</i>	Overfow

4.2 Gegevens lijn 2

<i>Van:</i>	Control
<i>Naar:</i>	ALU
<i>Bit lijnen:</i>	4
<i>Betekenis:</i>	ALU aansturing
<i>0000</i>	Pass S1
<i>0001</i>	Pass S2
<i>0010</i>	ADD
<i>0011</i>	NOR
<i>0100</i>	OR
<i>0101</i>	SLL
<i>0110</i>	SLTI
<i>0111</i>	SLTU
<i>1000</i>	SRA
<i>1001</i>	SRA
<i>1010</i>	SRL
<i>1011</i>	SRL
<i>1100</i>	SSL
<i>1101</i>	SUB
<i>1110</i>	XOR
<i>1111</i>	Empty

4.3 Gegevens lijn 3

<i>Van:</i>	Control
<i>Naar:</i>	Cache C
<i>Bit lijnen:</i>	1
<i>Betekenis:</i>	Laad gegevens in cache
<i>0</i>	Laad geen gegevens
<i>1</i>	Laad gegevens

4.4 Gegevens lijn 4

<i>Van:</i>	Control
<i>Naar:</i>	Register
<i>Bit lijnen:</i>	15
<i>Betekenis:</i>	Aanbieden register gegevens
<i>Formaat:</i>	XXXXXYYYYYYZZZZZ
<i>XXXXX</i>	Zet de waarde van register adres XXXXX op port a
<i>YYYYY</i>	Zet de waarde van register adres YYYYY op port b
<i>ZZZZZ</i>	Schrijf de waarde van port c in register adres ZZZZZ

4.5 Gegevens lijn 5, 6, 7, 8, 9, 10

<i>Van:</i>	Control
<i>Naar:</i>	Cache
<i>Bit lijnen:</i>	3
<i>Betekenis:</i>	Gegevens doorvoeren of niet
<i>Formaat:</i>	XYZ
<i>X</i>	Laad waarde in cache (aan/uit)
<i>Y</i>	Schrijf waarde naar bus S1 (aan/uit)
<i>Z</i>	Schrijf waarde naar bus S2 (aan/uit)

4.6 Gegevens lijn 11, 16

Van:	Control
Naar:	MUX
Bit lijnen:	1
Betekenis:	Selecteer welke data lijn doorgegeven moet worden
0	Selecteer a
1	Selecteer b

4.7 Gegevens lijn 12

Van:	Memory
Naar:	Control
Bit lijnen:	1
Betekenis:	Controleer of Memory bezig is met verwerken
0	Laad geen gegevens
1	Laad gegevens

4.8 Gegevens lijn 13

Van:	Control	
Naar:	Memory	
Bit lijnen:	4	
Betekenis:	Lees en schrijf naar geheugen	
Formaat:	XXYY	
Code	XX	YY
00	Doe niets	Byte
01	Lees	Half Word
10	Schrijf	Word
11	N.A.	Leeg

4.9 Gegevens lijn 11, 16

<i>Van:</i>	Control
<i>Naar:</i>	SRAS
<i>Bit lijnen:</i>	7
<i>Betekenis:</i>	Shift Right Arametic Special
<i>Formaat:</i>	XXYYYYYY
<i>YYYYY:</i>	Aantal bits dat geshift moet worden
<i>XX</i>	
<i>00</i>	Signextend
<i>01</i>	N.A.
<i>10</i>	Gebruik 0 als signextend
<i>11</i>	Gebruik 1 als signextend

4.10 Gegevens lijn 15

<i>Van:</i>	Control
<i>Naar:</i>	SRAS
<i>Bit lijnen:</i>	5
<i>Betekenis:</i>	Shift Left Logical
<i>XXXXX</i>	Aantal bits dat naar links geshift moet worden

4.11 Gegevens lijn 17

<i>Van:</i>	Control
<i>Naar:</i>	MUX
<i>Bit lijnen:</i>	5
<i>Betekenis:</i>	Invoer van getal

5 Microcode

5.1 Gedeelte 1

LOC	LABEL	DEST.	ALU	S1	S2	C	MISCELANNEOUS	CONDITION	JUMP
1	lfetch:							Interrupt	
2	lloop:						Instruction Read	Mem?	lloop
3		PC	ADD	PC	Constant	4	AB<-RF		DecodeMain
4	MemStore:	MAR	ADD	A	Imm32				
5		MDR	Pass S2		B				DecodeStore
6	SB:						Data Write + signbox	Mem?	lfetch
7	SH:						Data Write + signbox	Mem?	lfetch
8	SW:						Data Write + signbox	Mem?	lfetch
9	MemLoad:	MAR	ADD	A	Imm32				DecodeLoad
10	LB:	C	Pass S2				Date read, + signbox	Mem?	Read1
11	LBU:	C	Pass S2				Date read, + signbox	Mem?	Read1
12	LH:	C	Pass S2				Date read, + signbox	Mem?	Read1
13	LHU:	C	Pass S2				Date read, + signbox	Mem?	Read1
14	LW:	C	Pass S2				Date read, + signbox	Mem?	Read1
15	Read1:						Rt<-C		lfetch
16	BEQ:		SUB	A	B			0?	Branch
17								Uncond	lfetch
18	BNE:		SUB	A	B			0?	lfetch
19	Branch:	PC	ADD	PC	Imm32			Uncond	lfetch
20	J:	PC	ADD	PC	Imm32			Uncond	lfetch
21	JAL:	C	Pass S1	PC					
22		PC	ADD	PC	Imm32		R31<-C	Uncond	lfetch
23	Reglnit:								DecodeReg
24	SLL:	C	SSL	A	Imm32			Uncond	Write
25	SRL:	C	SRL	A	Imm32			Uncond	Write
26	SRA:	C	SRA	A	Imm32			Uncond	Write
27	SLLV:	C	SLL	A	B			Uncond	Write
28	SRLV:	C	SRL	A	B			Uncond	Write
29	SRAV:	C	SRA	A	B			Uncond	Write

5.2 Gedeelte 2

LOC	LABEL	DEST.	ALU	S1	S2	C	MISCELANNEOUS	CONDITION	JUMP
30	JALR:	C	Pass S1	PC					
31							Rspec<-C		
32	JR:	PC	Pass S1	A				Uncond	lfetch
33	ADD:	C	ADD	A	B			Overflow?	lfetch
34									Write
35	ADDU:	C	ADD	A	B			Uncond	Write
36	SUB:	C	SUB	A	B			Overflow?	lfetch
37									Write
38	SUBU:	C	SUB	A	B			Uncond	Write
39	AND:	C	AND	A	B			Uncond	Write
40	OR:	C	OR	A	B			Uncond	Write
41	XOR:	C	XOR	A	B			Uncond	Write
42	NOR:	C	NOR	A	B			Uncond	Write
43	SLT:	C	SLT	A	B			Uncond	Write
44	SLTU:	C	SLTU	A	B			Uncond	Write
45	Imm:	Temp	Pass S2		Imm32			DecodImm	
46	ADDIU:	C	ADD	A	Temp			Uncond	Write
47	SLTI:	C	SLTI	A	Temp			Uncond	Write
48	SLTIU:	C	SLTIU	A	Temp			Uncond	Write
49	ANDI	C	AND	A	Temp			Uncond	Write
50	ORI:	C	OR	A	Temp			Uncond	Write
51	XORI:	C	XOR	A	Temp			Uncond	Write
52	LUI:	C	Pass S2		Imm32			Uncond	Write
53	ADDI:	C	ADD	A	Temp			Overflow?	lfetch
54	Write:						Rd<-C	Uncond	lfetch