Technische Grundlagen der Informatik 2

Rechnerorganisation

Probeklausur
(Klausur September 2010)

Prof. Dr. Ben Juurlink

Fachgebiet: Architektur eingebetteter Systeme
Institut für Technische Informatik und Mikroelektronik
Fak. IV – Elektrotechnik und Informatik

SS 2014

Aufgabe 1: MIPS Assemblersprache

- Übersetzen Sie die folgende Funktion nach MIPS-Assembler.
- Beachten Sie dabei die MIPS-Register Konventionen.
- Pseudo-Instruktionen dürfen verwendet werden.

```
void clip(int a[], int n)
{
   int i;

   for (i=0; i<n; i++)
   {
      if (a[i]<0)
         a[i]=0;
      else if (a[i]>255)
         a[i] = 255;
   }
}
```

Aufgabe 1: Lösungsblatt

```
void clip(int a[], int n){
   int i;

   for (i=0; i<n; i++)
      if (a[i]<0)
         a[i]=0;
      else if (a[i]>255)
         a[i] = 255;
}
```

Aufgabe 2: Lösungsblatt

- Stellen Sie die Zahl 19,75 in binärer Darstellung nach IEEE 754 mit einfacher Genauigkeit dar.
  - $19_{10} = 2^4 + 2^1 + 2^0 = 10011_2$
  - $0{,}75_{10} = 2^{-1} + 2^{-2} = .11$
  - $19{,}75_{10} = 10011.11_2 = 1.001111 * 2^4$
  - S=0, da positiv
  - e = E + Bias = 4 + 127 = $131_{10} = 10000011_2$
  - → 0 | 10000011 | 00111100000000000000000

Aufgabe 3: Maschinensprache

- Stellen Sie folgenden Code in MIPS-Maschinencode (dezimal) dar:
  - Schleife beginnt an Adresse 1000
  - Verwenden Sie „MIPS Reference Card“
  - Beispiel Darstellung:

1000	1	2	3	4	5	6

```
L: sll   $t1,$t0,2
   add   $t1,$a0,$t1
   lw    $t1,-4($t1)
   beq   $t1,$a1,E
   addi  $t0,$t0,1
   j     L
E:  …
```

Aufgabe 3: Maschinensprache Lösung

- Stellen folgenden Code in MIPS-Maschinencode (dezimal) dar:
  - Schleife beginnt an Adresse 1000

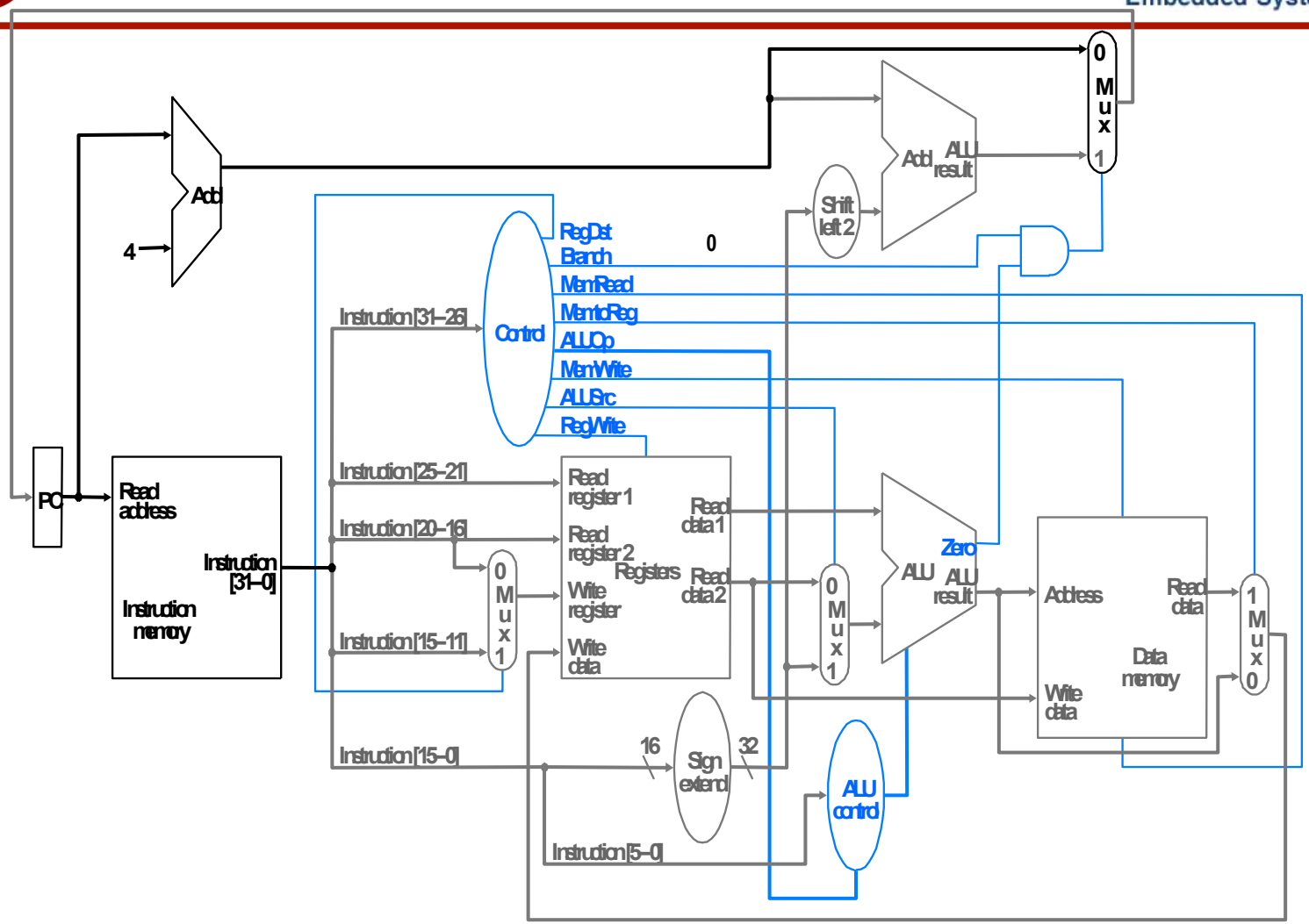
1000	0	0	8	9	2	0	`L: sll $t1,$t0,2`
1004	0	4	9	9	0	32	`add $t1,$a0,$t1`
1008	35	9	9	-4			`lw $t1,-4($t1)`
1012	4	9	5	2			`beq $t1,$a1,E`
1016	8	8	8	1			`addi $t0,$t0,1`
1020	2	250					`j L`
1024	...						`E:` …

Aufgabe 4: Der Eintaktprozessor

- Wir müssen den Eintaktprozessor um den Befehl **`ori`** (OR-immediate) erweitern.
  - Ergänzen Sie benötigte Datenpfade und Steuersignale in der Abbildung auf der nächsten Folie
  - Geben Sie die Werte an, die die Steuersignale haben müssen, so dass der Datenpfad den ori-Befehl ausführt. Verwenden Sie falls möglich Don't Cares. Erweitern sie ggf. die ALU.
- **`ori`** Befehlsformat:

I-Format	opcode	rs	rt	imm
ori rt,rs,imm	0xd	rs	rt	imm

Aufgabe 4: Lösungsblatt (1)



Aufgabe 4: Lösungsblatt (2)

- Steuersignale:

RegDst	Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite

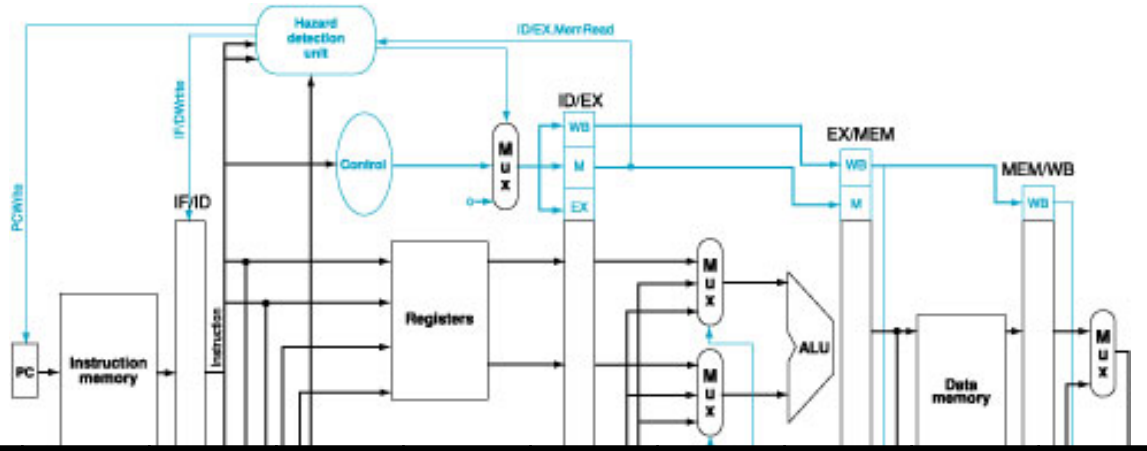
- *ALU Erweiterung:

ALUOp	Funct-Feld	ALU-Aktion	ALU-Steuereingang

Aufgabe 5: Pipelining

- Gegeben folgender Assemblercode
- Wird auf dem 5-Stufen-Pipeline-MIPS ausgeführt mit vollständiger Forwarding Hardware → Wenn geforwardet werden *kann*, wird geforwardet
- Vervollständigen Sie das Pipeline-Diagramm
  - Für jede Instruktion Kürzel IF, ID, EX, MEM, WB eintragen.
  - Wartezyklen(*stall cycles*) kennzeichnen mit X
  - Forwards kennzeichnen mit Pfeil (→) zwischen beteiligten Stufen

```
I0: add $4,$1,$0
I1: sub $9,$3,$4
I2: add $5,$4,$7
I3: lw  $2,100($3)
I4: lw  $2,0($2)
I5: and $2,$2,$1
I6: beq $9,$1,target
```



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
I0														
I1														
...														

Aufgabe 5: Pipelining - Lösungsblatt


```
I0: add   $4,$1,$0
I1: sub   $9,$3,$4
I2: add   $5,$4,$7
I3: lw    $2,100($3)
I4: lw    $2,0($2)
I5: and   $2,$2,$1
I6: beq   $9,$1,target
I7: and   $9,$9,$1
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
I0														
I1														
I2														
I3														
I4														
I5														
I6														
I7														

Aufgabe 6: Caches - Lösung

- Gegeben ist ein Cache mit folgenden Daten:
  - 8-fach assoziativ
  - 64 Blöcke mit jeweils 32 B
  - Adresslänge: 32 bit
- Wieviele Bits werden für den Index, Offset und Tag benötigt?
  - 64 Blöcke sowie 8-fach assoziativ → 8 Sätze → 3 bit
  - Offset: 3 bit + 2 bit = 5 bit (Wort- + Byteoffset)
  - Tag = 32 bit – (3 bit + 5 bit) = 24 bit
- Was ist ein „Dirty Bit“? Bei welcher Schreibstrategie wird es verwendet?
  - Zeigt an, ob der Block verändert (geschrieben) wurde.
  - Write-Back-Strategie / Rückscheibestrategie