

4. Übungsblatt Informatik II

(Abgabe: 18.05.2005)

1. Aufgabe:

Zu berechnen sind aus $I(c)[31:0]$ die Prädikate

- $RD(c)$
- $sxtimm(c)$
- $jump(c)$
- $branch(c)$
- $lw(c)$
- $sw(c)$
- $comp(c)$
- $compimm(c)$
- $jal(c)$
- $jalr(c)$
- $RS1(c)$
- $RS2(c)$
- $aluf(c)$

Auf die Schaltkreis-Darstellung soll hier verzichtet werden, stattdessen zu jedem Prädikat eine Berechnungsvorschrift. Die Steuersignale $rtype$, $itype$ und $jtype$ werden nachstehend berechnet, ebenso wird eine einfache Möglichkeit angegeben, die Instruktion auf Bit-Gleichheit mit einem vorgegebenen Muster (laut Instruktionssatz) zu testen (z.B. $=10011$).

- $RD(c)$ - wird direkt aus der Instruktion dekodiert.
Fallunterscheidung mit Hilfe eines MUXes, am Ausgang die ersten 5 Bits abgreifen.
 - (a) $rtype = 1: I[15:11]0^{26}$
 - (b) $sonst: I[20:16]0^{26}$
- $sxtimm(c)$ - Immediate-Konstante wird direkt aus der Instruktion sign-extended.
Fallunterscheidung durch einen MUX, als Eingänge die jeweils Vorzeichenerweiterte immediate-Konstante.
 - (a) $jtype = 1: I[25]^6 I[25:0]$
 - (b) $sonst: I[15]^{16} I[15:0]$
- $jump(c)$ - $jalr \vee jr \vee j \vee jal$.
- $branch(c)$ - $I[31:27]=11010$.
- $lw(c)$ - $I[31,30,28:26]=10011 \wedge itype \wedge !I[29]$.
- $sw(c)$ - $I[31,30,28:26]=10011 \wedge itype \wedge I[29]$.
- $comp(c)$ - $NOR(I[5],I[4]) \wedge rtype$.
- $compimm(c)$ - $!I[31] \wedge I[30] \wedge itype$.
- $jal(c)$ - $I[31:27]=00001 \wedge jtype \wedge I[26]$.
- wg. $jump: j(c)$ - $I[31:27]=00001 \wedge jtype \wedge !I[26]$.
- $jalr$ - $I[31:27]=11011 \wedge itype \wedge I[26]$.

4. Übungsblatt Informatik II

(Abgabe: 18.05.2005)

- wg. jump: jr - I[31:27]=11011 \wedge itype \wedge !I[26].
- RS1(c) - I[25:21].
- RS2(c) - I[20:16].

Um die I[n:n-4]=abcde - Operation zu berechnen, ver-exor-t man jeweils die 5 Bits I[n:n-4] mit den 5 Bits abcde bitweise. Ein NOR aller dieser 5 Bits ergibt bei Gleichheit 1, sonst 0. In der Vorlesung wurden jtype,rtype und itype definiert als

- itype: I[31:27]=00001 \vee I[31:27]=11111
- rtype: I[31:27]=00000 \wedge !I[26]
- jtype: litype \wedge lrtype

2. Aufgabe:

Sei c[31:0] die zu ladende 32-bit Konstante.

Da die Immediate-Konstante sign-extended wird, muss der Compiler eine Fallunterscheidung abhängig von Bit c[15] treffen.

1. Fall: c[15] = 0

lhgi (R1, xxxxx, c[31:16])
xori (R2, R1, c[15:0])

2. Fall: c[15] = 1

lhgi (R1, xxxxx, $\overline{c[31:16]}$)
xori (R2, R1, c[15:0])

3. Aufgabe: a.)

0:addi(R1, R0, 100); R1:=100 (lädt 100 in R1 da R0=0)
4:lw(R2, R1, 0); R2:=a₁ (da R1=100 und a₁ auf Adresse 100 steht)
8:lw(R3, R1, 4); R3:=a₂ (da R1+4 = 104 und a₂ auf 104 steht)
12:add(R1, R2, R3); R1:=a₁ + a₂ (da R2=a₁ und R3=a₂, speichert R1 a₁ + a₂)
16:sw(R1, R0, 1024) (speichert Ergebnis in Adresse 1024)

b.)

Ausser den Adressen die am Anfang geladen werden müssen ändert sich nichts da die Adressen auch weiterhin kleiner $2^{15} - 1$.

c.)

0:lw(R1, R0, 200);R1:=[203:200] (speichert n in R1)
4:add(R2, R1, R0);R2:=R1 (kopieren von n in R2)
8:beqz(R1, 16);PC:=PC+16=24 (Schleifenende?)
12:subi(R1, R1, 1);R1:=R1-1 (n wird runtergezählt)

4. Übungsblatt Informatik II

(Abgabe: 18.05.2005)

16:add(R2, R2, R1);R2:=R2+R1

20:j(-12);PC=PC-12=8 (zurück zum Anfang der Schleife)

24:sw(R2, R0, 1024); n[1027:1024] = R2 (abspeichern)

4. Aufgabe: Misalignment

Sei $a \in \{0, 1\}^n$ also $a[n-1:0]$ die Adresse des Speicherzugriffs. In der Binärcodierung ist a dann durch 4 teilbar, wenn die letzten beiden Stellen Null sind.

$$\text{misalignment}(a) = \begin{cases} 1 & : a \text{ nicht durch 4 teilbar} \\ 0 & : \text{sonst} \end{cases}$$

$$\Rightarrow \text{misalignment}(a) = a_0 \wedge a_1$$

5. Aufgabe: Illegal

$Ill : \{0, 1\}^{32} \rightarrow \{0, 1\}$ mit

$$Ill(I(c)) = \begin{cases} 0 & : I(c) \text{ ist gültige Instruktion} \\ 1 & : \text{sonst} \end{cases}$$

$$\begin{aligned} \overline{Ill(I(c))} &= \underbrace{\left(\overline{I[31:26]} \wedge \underbrace{\left(\underbrace{\overline{I[5:4]}}_{\text{compute}} \vee \underbrace{\overline{I[5]I[4]I[3:1]}}_{\text{specialmoves}} \right)} \right)}_{R\text{-Type}} \vee \underbrace{\left(\overline{I[31:27]} \vee \overline{I[31:28]I[27]} \right)}_{J\text{-Type}} \\ &\vee \underbrace{\left(\overline{I[31]I[30]} \vee \underbrace{\left(\overline{I[31:30]I[29]I[28]} \right)}_{\text{control}} \vee \underbrace{\left(\overline{I[31]I[30]I[28]I[27:26]} \right)}_{\text{load/store}} \right)}_{I\text{-Type}} \end{aligned}$$