

Computer Architecture 1 - Übungsblatt 5

Dirk Heine, 2030941
Jan Hendrik Dithmar, 2031259
Steffen Heil, 2019358

Aufgabe 1

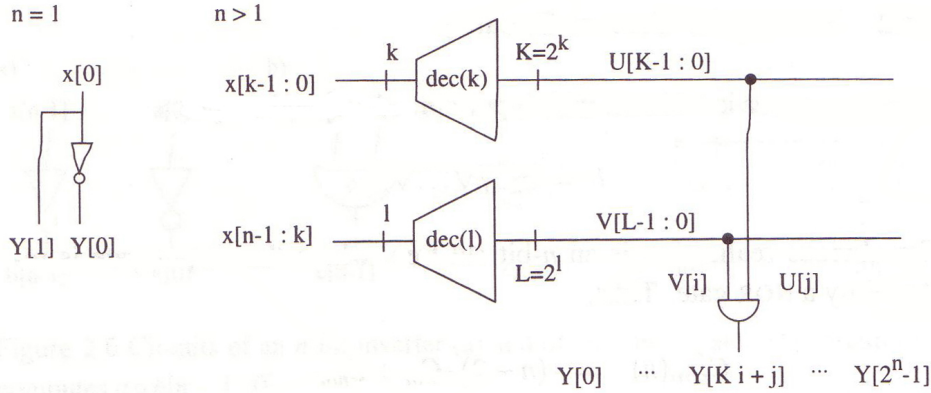


Abbildung 1: n -Decoder

- $n = 1$:

$$Y_0 = \neg x_0$$

$$Y_1 = x_0$$

- $n > 1$:

$$m := \lceil n/2 \rceil$$

$$k := n - m$$

$$A[2^k - 1 : 0] := \text{dec}_k(x[n-1 : m])$$

$$B[2^m - 1 : 0] := \text{dec}_m(x[m-1 : 0])$$

$$\forall a \in \{0, \dots, 2^k - 1\}, \forall b \in \{0, \dots, 2^m - 1\}:$$

$$Y_{a \cdot 2^m + b} := A_a \wedge B_b$$

Proof:

- $n = 1$: trivial

- $n > 1$:

$$\begin{aligned} a &:= \left\lfloor \frac{i}{2^m} \right\rfloor \\ b &:= i - a \cdot 2^m \\ \Rightarrow i &= a \cdot 2^m + b \end{aligned}$$

$$\begin{aligned} Y_i = 1 &\stackrel{def.a,b}{\Leftrightarrow} Y_{a \cdot 2^m + b} \\ &\Leftrightarrow A_a = 1 \wedge B_b = 1 \\ &\Leftrightarrow dec_k(x[n-1:m])_a = 1 \wedge dec_k(x[m-1:0])_b = 1 \\ &\stackrel{IA}{\Leftrightarrow} \langle x[n-1:m] \rangle = a \wedge \langle x[m-1:0] \rangle = b \\ &\Leftrightarrow \langle x[n-1:0] \rangle = a \cdot 2^m + b \\ &\stackrel{def.a,b}{\Leftrightarrow} \langle x \rangle = i \end{aligned}$$

Aufgabe 3

Notation:

```
mnemonic rd rs1 rs2
mnemonic rd rs1 imm
mnemonic rs1 label
```

Sequentiell:

```
*
* Grundannahme: Mindestens ein Wert! (Sonst kein Maximum...)
*
* R1: Anzahl der Werte
* R2: Adresse des zuletzt gelesenen Wertes
*      zugleich Anzahl der noch zu lesenden Werte
* R3: Bisheriges Maximum
* R4: Zuletzt gelesener Werte (außer erster)
*
* Wir durchlaufen das Feld rückwärts, damit die Adresse dem
* Zähler entspricht.
* Wir sparen dadurch eine Register bzw. einen Vergleich.
*
      subi  r2 r1 1      * Adresse des letzten Wertes
      lw    r3 r1 -1     * letzten Wert als initial-Maximum
check: beqz  r2 finish    * stehen noch Werte aus?
next:  subi  r2 r2 1     * nächste Adresse
      lw    r4 r2 0      * nächster Wert
      sge   r5 r3 r4     * Ist das alte Maximum größer
                        * oder gleich ?
      bnez  r5 check     * ja => unverändert lassen
                        * (nächsten Wert holen)
      addi  r3 r4 0      * nein => neues Maximum merken
      bnez  r2 next      * Weitere Werte?
finish: sw   r3 r1 0     * Ergebniss speichern
```

Aufgabe 4a

Mit delayed branch:

```
*
* Grundannahme: Mindestens ein Wert! (Sonst kein Maximum...)
*
* R1: Anzahl der Werte
* R2: Adresse des zuletzt gelesenen Wertes
*      zugleich Anzahl der noch zu lesenden Werte
* R3: Bisheriges Maximum
* R4: Zuletzt gelesener Werte (außer erster)
*
* Wir durchlaufen das Feld rückwärts, damit die Adresse dem
* Zähler entspricht.
* Wir sparen dadurch eine Register bzw. einen Vergleich.
*
      subi  r2 r1 1      * Adresse des letzten Wertes
      lw    r3 r2 0      * letzten Wert als initial-Maximum
check:  beqz  r2 finish   * stehen noch Werte aus?
next:   subi  r2 r2 1     * nächste Adresse
      lw    r4 r2 0      * nächster Wert
      sge   r5 r3 r4      * Ist das alte Maximum größer
                        * oder gleich ?
      bnez  r5 check      * ja => unverändert lassen
                        * (nächsten Wert holen)
      bnez  r2 next       * Weitere Werte?
      addi  r3 r4 0       * nein => neues Maximum merken
finish: sw    r3 r1 0     * Ergebniss speichern
```

Aufgabe 4b

```
        beqz r0 db
dpc:    beqz r0 +4
        <command2>
        ...
db:     <command1>
        <command3>
        ...
```

Evaluation order:

Delayed PC:

1. beqz r0 db
2. beqz r0 +4
3. <command1>
(when executing 2. the address for 4. is
calculated based on the delayed pc to be dpc + 4.)
4. <command2>
5. ...

Delayed Branch:

1. beqz r0 db
2. beqz r0 +4
3. <command1>
(when executing 2. the address for 4. is
calculated based on the current pc to be db + 4.)
4. <command3>
5.

Aufgabe 5

Wir definieren:

$$\begin{aligned} GPRr(c, x) &:= itype \wedge c.IR[20 : 16] = x \\ &\quad \vee rtype \wedge (c.IR[20 : 16] = x \vee c.IR[15 : 11] = x) \\ &\quad \vee itype \wedge c.IR[31 : 29] = 101 \wedge c.IR[25 : 21] = x \\ GPRw(c, x) &:= itype \wedge c.IR[31 : 29] <> 101 \wedge c.IR[25 : 21] = x \wedge c.IR[31, 29 : 28] = 001 \\ &\quad \vee rtype \wedge c.IR[25 : 21] = x \end{aligned}$$

$$SC6 : \forall i. \forall x \in \{0 \dots 31\}. \forall n \in \{1 \dots 3\}. GPRr(c^i, x) \Rightarrow \neg GPRw(c^{i-n}, x)$$