



Computer Architecture I - WS 02/03  
(due: 09.12.2002)

---

**Aufgabe 1: (Inkrementer)** ( 2 + 2 + 3 Punkte)

Ein Inkrementer ist ein Schaltkreis der aus den Eingängen  $x \in \{0, 1\}^n$  und  $c \in \{0, 1\}$  die Ausgabe  $y \in \{0, 1\}^{n+1}$  mit  $\langle y \rangle = \langle x \rangle + \langle c \rangle$  berechnet.

1. Konstruieren Sie einen Inkrementer nach dem Vorbild eines Carry Chain Addierers.
2. Geben Sie eine geschlossene Formel für Kosten und Delay ihrer Konstruktion an. Beweisen Sie die Korrektheit ihrer Formeln.
3. Konstruieren Sie einen Inkrementer nach dem Vorbild eines Conditional Sum Addierers. Berechnen Sie Kosten und Delay ihrer Konstruktion.

**Aufgabe 2: (Programmzähler)** ( 1 + 2 + 2 Punkte)

Ein  $n$  Bit Zähler ist ein Schaltkreis mit Eingängen  $clk$  und  $reset$  und der Ausgabe  $y \in \{0, 1\}^n$ . Die Ausgabe  $y$  von Zyklus  $t + 1$  wird berechnet durch:

Falls  $reset = 0$  dann ist  $\langle y^{t+1} \rangle = \langle y^t \rangle + 1 \pmod{2^n}$  ansonsten  $y^{t+1} = 0^n$ .<sup>1</sup>

1. Konstruieren Sie einen  $n$  Bit Zähler. Benutzen Sie dazu einen  $n$  Bit Inkrementer und ein  $n$  Bit Register. Das Register hat einen zusätzlichen Reset-Eingang, mit dem der gespeicherte Wert auf  $0^n$  gesetzt werden kann.
2. Konstruieren Sie einen Programmzähler (PC) für die DLX. Abhängig vom Eingangssignal  $s$  soll Ihr Zähler um vier erhöht werden ( $s = 0$ ) oder auf  $0^n$  gesetzt werden ( $s = 1$ ). Benutzen Sie einen Inkrementer zur Konstruktion ihres Zählers.
3. Beweisen Sie, dass Ihre Konstruktion den Programmzähler korrekt um vier erhöht, falls  $s = 1$ .

**Aufgabe 3: (Software Multiplikation)** ( 7 Punkte)

Der DLX Prozessor hat keine Instruktion für Integer Multiplikation. Schreiben Sie ein Assemblerprogramm, dass die Zahlen in Register  $GPR(1)$  und  $GPR(2)$  miteinander multipliziert. Das Ergebnis soll ein Register  $GPR(3)$  gespeichert werden.

**Aufgabe 4: (Adressen)** ( 2 Punkte)

Da alle Berechnungen in der DLX modulo  $2^{32}$  durchgeführt werden, ist es egal ob Adressen als binäre Zahlen oder Zweierkomplementzahlen aufgefasst werden. Der Grund dafür ist, dass die folgende Gleichung für 32 Bit Adressen  $a$  gilt:

$$[a] \equiv \langle a \rangle \pmod{2^{32}}$$

Beweisen Sie die Korrektheit dieser Gleichung.

---

<sup>1</sup> $y^t$  ist der Wert der Ausgabe  $y$  des Counters nach  $t$  Taktzyklen.

**Aufgabe 4: (Von Neumann Addierer)**

**( 4 + 3 + 2 Punkte)**

In Abbildung 1 sehen Sie die Konstruktion eines so genannten Von Neumann Addierers. Er ist aus den beiden Registern  $Y$  und  $X$  und  $n + 1$  vielen Halbaddierern aufgebaut. Für natürliche Zahlen  $t$  gibt  $X^t$  und  $Y^t$  den Inhalt der Register nach  $t$  Taktzyklen an.  $X^0$  und  $Y^0$  sind also die initialen Werte.

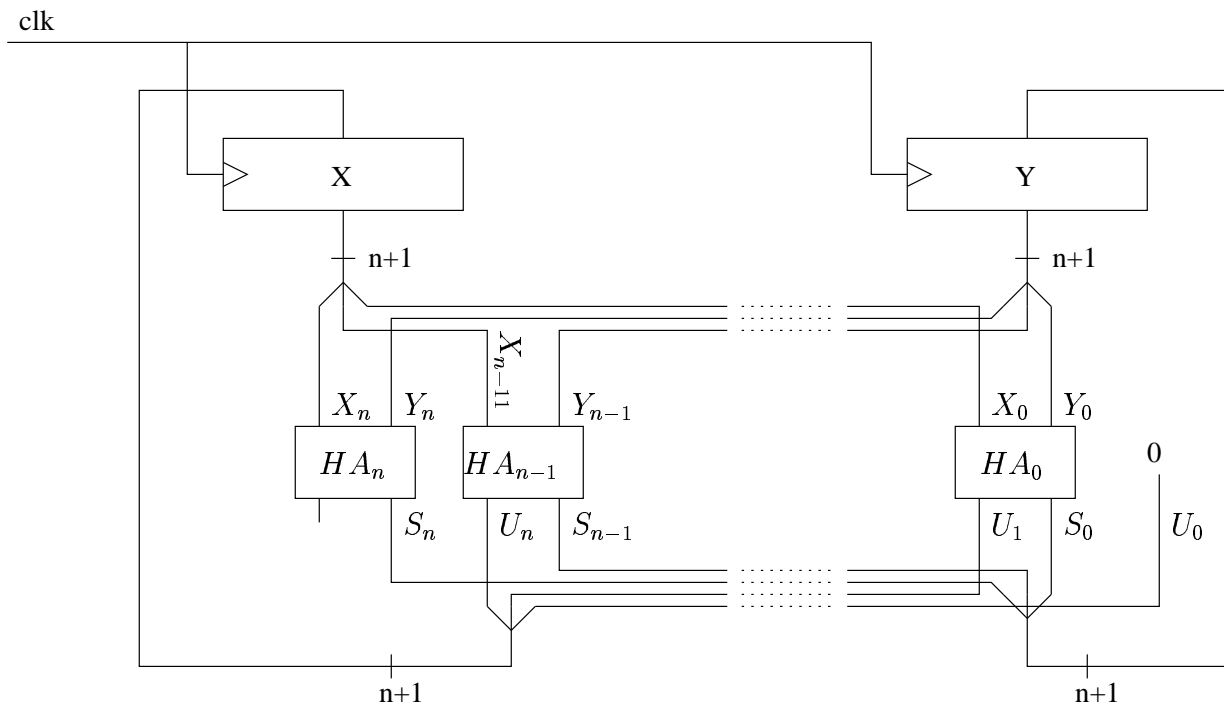


Figure 1: Von Neumann Addierer

1. Seien die höchstwertigsten Bits der beiden Register mit 0 initialisiert:  $X_0^n = 0$  und  $Y_0^n = 0$ . Zeigen Sie die Korrektheit des folgenden Lemmas :  $\langle X^t \rangle + \langle Y^t \rangle = \langle X^0 \rangle + \langle Y^0 \rangle$  für alle  $t \geq 0$ .
2. Beweisen Sie:  $\exists t \leq n + 1 : \langle Y^t \rangle = \langle X^0 \rangle + \langle Y^0 \rangle$ . Das beweist, dass der Von Neumann Addierer maximal  $n + 1$  Taktzyklen braucht, um die Summe von  $X^0$  und  $Y^0$  zu berechnen und im Register  $Y$  zu speichern.
3. In welchem Fall braucht dieser Addierer genau  $n + 1$  Takte für die Addition?