

## ITS Teil 2: Rechnerarchitektur

### 1. Grundsaltungen der Digitaltechnik

- a. Flipflop (taktflankengesteuert) → Wdh. Signalverläufe beim D-FF
- b. Zähler (Bsp. 4-Bit Zähler)
  - Eingang „count“ wird zum Aktivieren der Zählung genutzt (count = 1 → Zähler zählt mit den Taktflanken weiter, count = 0 → Zähler hält den aktuellen Wert)
  - Eingang  $\overline{CLR}$  setzt den Zähler bei Aktivierung wieder auf 0
  - Leitungen werden als Busse zusammengefasst (statt 4 Leitungen zu zeichnen, nutzt man eine dicker gezeichnete Leitung, an der die Anzahl der enthaltenen Leitungen notiert wird)
  - Im Signalverlauf wird der Wert des Busses genannt, nicht jener der einzelnen Binärstellen
  - Der „count“-Eingang wird mit dem CLK (Takteingang → Clock) auf ein AND-Gatter geführt, sodass nur dann der Takt weitergereicht wird, wenn „count“ den Wert 1 hat
  - Anwendung: Der Zähler ist im Prozessor integriert und arbeitet dort als Befehlszähler (Adresse des nächsten Befehls → Program Counter)
- c. Asynchrnzähler
  - Ohne zusätzliche Logik → besteht nur aus gekoppelten D-FFs
  - Sollte eine Taktflanke ankommen, so wechselt das FF jeweils den Zustand zwischen 1 und 0 hin und her
    - Das erste FF hat eine Wechselfrequenz, die der halben Taktfrequenz entspricht, das zweite hat entsprechend eine Frequenz, die einem Viertel der Taktfrequenz entspricht → man bezeichnet den Zähler als asynchron
  - Betrachtet man die Zustände an den Ausgängen der FFs und ordnet diesen Binärstellen zu (1. FF →  $2^0$ , 2. FF →  $2^1$ , usw.) entsteht beim Auswerten ein hochlaufender Zähler
- d. Register
  - Dient zur Speicherung von n-Bits → einzelne FFs, die zusammengefasst sind und somit entsprechende Anzahlen von Werten einer Binärzahl speichern können
  - FFs haben alle identischen Takt
  - Bsp. Zahl 65 in einem 8-Bit Register:  
 $2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$   
0 1 0 0 0 0 0 1
- e. Arithmetisch logische Einheit (ALU = arithmetical logical unit)
  - Führt Rechenoperationen im Prozessor durch (→ auch logische Verknüpfungen)
  - Per Binärkode wird die gewünschte Operation gewählt
  - Sonderfälle beim Ergebnis werden durch Flags ausgegeben (Ergebnis = 0, Ergebnis < 0)

#### f. Multiplexer (MUX)

- Bietet die Möglichkeit, einen Eingang aus mehreren Eingängen auf einen Ausgang zu Schalten
- Besitzt neben Eingängen und Ausgang noch Steuereingänge (Select) und einen Einschalter (Enable)
- Anwendung: Die ALU berechnet jeweils alle Ergebnisse von verschiedenen enthaltenen Rechenoperationen, ein Multiplexer gibt nur ein Ergebnis weiter, welches zuvor per Select ausgewählt wurde

#### g. Bussystem

- Bussysteme sind nur Transportmechanismen (im einfachsten Fall sind es Leitungen)
- An den Bus werden verschiedene Schaltungsbausteine angeschlossen
  - ➔ es dürfen nicht zwei verschiedene Zustände an den Bus angeschlossen sein
  - ➔ Tristate Ausgänge ermöglichen, einen Ausgang „abzuschalten“ sodass dieser keinen Wert mehr auf den Bus legen kann
- Bsp.: Eine 3 aus Register A soll auf den Bus gelegt werden ➔ es werden die Tristate Ausgänge von Register A geschaltet und so die Zustände auf den Bus übertragen
  - ➔ es dürfen nicht Ausgänge verschiedener Schaltungselemente gleichzeitig aktiv sein
- Steuersignal für Tristats ➔  $O_E$  (Output Enable)
- Damit die Tristats funktionieren, müssen alle Elemente denselben Takt erhalten
- Bsp. für Datentransfer:
  - Einschalten der Tristate-Ausgänge für ein Register A ( $E_A = 1$ , Enable A)
  - Spannungen liegen auf Bus an
  - Auf Register B werden die Eingänge geschaltet ( $L_B = 1$ , Load B), sodass bei der nächsten Taktflanke der Inhalt des Busses (also jener aus Register A) in Register B geladen wird ( $E_A$  &  $L_B$  müssen bei steigender CLK-Flanke aktiv sein, damit der Transfer funktioniert)

## 2. Rechnerarchitektur DemoCom

- 16 Byte RAM ➔ 4 Bit Adressgröße
- Betriebsmodi Programmieren & Ausführen:
  - Im Modus Programmieren werden von außen Daten in den Speicher oder das Adressregister geschrieben
  - Im Modus Ausführen werden Programme im Prozessor ausgeführt
- Schalter werden zur Eingabe von Werten im Programmiermodus genutzt
- Alle Stufen sind mit einem Tristate Bus verbunden

### Elemente des DemoCom

#### a. Programm-Counter:

- $C_p = count$  ➔ aktiviert den Zähler,  $\overline{CLR}$  ➔ setzt den Zähler zurück,  $E_p$  ➔ aktiviert den Ausgang des Zählers auf das Bussystem
- Es gibt einen definierten Startpunkt, an dem jeder Mikroprozessor sein Programm beginnt ➔ Speicherpunkt 0

b. Akkumulator:

- Eigentlich würden zur korrekten Berechnung von Werten 3 Register (2 Operatoren und 1 Ergebnis) benötigt
- Der Akkumulator enthält den Wert des Operant A und erhält nach der Berechnung den Wert des Ergebnisses
  - o Vorteil: Man spart ein Register
  - o Nachteil: Man kann den eigentlichen Wert des Akkumulators (Operant A) nach der Berechnung nicht weiter verwenden
- Sowohl D-Leitungen als auch Q-Leitungen des Akkumulators sind mit den Bus verbunden ( $L_A \rightarrow$  Load Akkumulator: Wert des Bussystems wird in den Akkumulator geladen,  $E_A \rightarrow$  Wert des Akkumulators wird ins Bussystem übertragen)

c. ALU

- $S_U \rightarrow$  Umschaltung Addition/Subtraktion ( $S_U = 1 \rightarrow$  Subtraktion aktiv)

d. Register B

- Enthält Operant B  $\rightarrow$  Q-Ausgänge des Registers sind nur auf ALU geführt
- Das Register wird vom Bus aus geladen

e. Output-Register

- Register, welches nur vom Bus geladen werden kann  $\rightarrow$  Ausgang wird auf Binary Display (LED-Anzeige) geführt
- LEDs geben den aktuellen Zustand aus (diese entsprechen den Binärstellen)

f. MUX

- Select ( $S_A$ )  $\rightarrow$  Wählt aus, ob vom Bus die Speicheradresse geladen werden soll (Modus Ausführen) oder ob vom externen Anschluss eine Adresse geladen werden soll (Modus Programmieren)
  - $\rightarrow$  Je 4 Bit werden weitergegeben an MAR

g. MAR (Memory Access Register)

- Bekommt seine 4-Bit vom MUX vorgegeben
- „hält“ die Speicheradresse, geladen wird es durch den Eingang  $L_M$

h. RAM (Random Access Memory)

- Speicher, welcher gelesen und geschrieben werden kann (Bsp. zu einer Variable wird etwas hinzuaddiert und unter demselben Variablennamen abgespeichert)
- Bsp. für „Nur-Lese“-Speicher  $\rightarrow$  BIOS am Computer  $\rightarrow$  Speicher muss erst in eine spezielle Betriebsart gebracht werden, damit eine Veränderung möglich ist
- Adresse wird vom MAR vorgegeben (16 Adressen  $\rightarrow$  MAR liefert 4 Bit)
- Daten werden von außen vorgegeben (Modus Programmieren,  $W_E = 1$ ) und auf den Bus geführt ( $C_E \rightarrow$  Tristate Ausgang)

i. Instruction Register

- 8 Bit werden gespeichert → 4 werden in die Ablaufsteuerung geführt, 4 werden auf den Bus geführt:

$$\begin{array}{cccccccc} I_7 & I_6 & I_5 & I_4 & I_3 & I_2 & I_1 & I_0 \\ \hline \text{Ablaufsteuerung} & & & & & & & \text{Bus} \end{array}$$

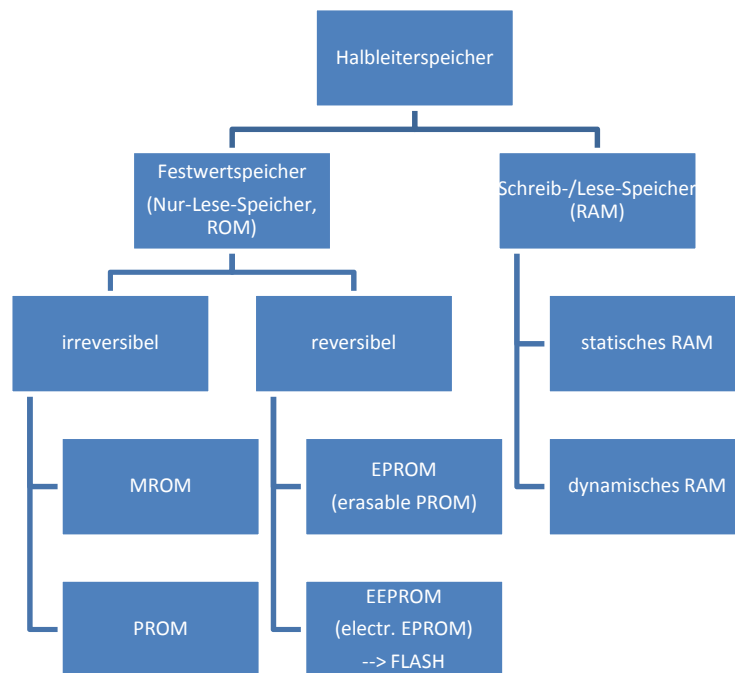
- Ansonsten ist das Instruction Register ein normales 8-Bit Register
- Register hält den aktuellen Befehl (welcher ausgeführt werden soll) → gibt diesen an die Ablaufsteuerung weiter

j. Ablaufsteuerung:

- Die „Intelligenz“ des Systems → sorgt dafür, dass die Befehle an die Elemente weitergegeben werden

Speicherbausteine

- „Gedächtnis“ eines Computers
- Es gibt verschiedene Ausführungen, sie unterscheiden sich in Arbeitsgeschwindigkeit, Dauer des Datenerhaltes, Technik des Löschsens und Beschreibens und der Organisation



- Gemeinsam ist allen Speicherbausteinen:
  - o Matrixartiger Aufbau durch Wort- und Bitleitungen
  - o An den Kreuzungspunkten von Wort- und Bitleitungen sitzen die Speicherzellen
  - o Der Aufbau der Speicherzellen ist technologieabhängig

- Der Inhalt der Speicherzellen zeigt sich, wenn die Wortleitung aktiviert wird (mit Spannungs-High-Pegel):
  - o Speicherzelle enthält eine 1 → Speicherzelle aktiviert die Bitleitung (High-Pegel)
  - o Speicherzelle enthält eine 0 → Speicherzelle aktiviert die Bitleitung nicht (Low-Pegel)
- Am Eingang wird die Adresse angegeben, z.B. können mit drei Adressleitungen je acht Wortleitungen gewählt werden
  - ➔ Die richtigen Speicherzellen (Wortleitung) werden selektiert, indem man die Adresse dekodiert (→ Adressdecoder)