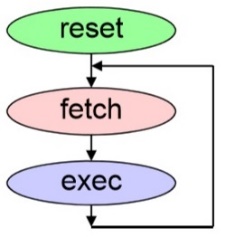
CT Zwischenprüfung 1

Moores Law: „Die Anzahl Transistoren wird sich pro Jahr verdoppeln“ – Real: 18 Monate

Recheneinheit: führt logische und arithmetische Befehle aus.  
Steuereinheit: liest, interpretiert und führt Instruktionen aus.

CPU – Central Processing Unit

* Recheneinheit:
  + ALU (Arithmetic and Logic Unit)
    - Führt Operationen aus (+, -, \*, /, …, &, #)
  + ACCU (Accumulator)
    - Register: Zwischenspeicher für Daten
  + Verarbeitungsbreite: 4/8/16/32/64 – bit
* Steuereinheit:
  + Finite State Machine (FSM)
    - Reset: Initialisierung, Startzustand
    - Fetch: Instruktionen lesen
    - Exec: Execute, Instruktionen ausführen

Memory – Arbeitsspeicher (Hauptspeicher)

* Menge von Speicherzellen
  + Eine Zelle := 8 Bit (1 Byte)
  + RAM (Random Access Memory): können gelesen und beschrieben werden.
* Zentraler Speicher
  + RAM -> flüchtig: nicht langzeitig
  + ROM (Read Only Memory) -> nicht flüchtig: in Fabrik vorprogrammiert

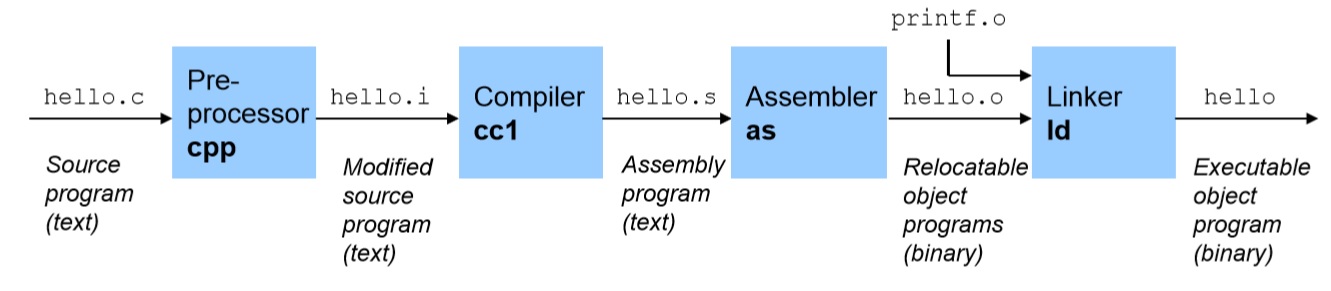
Memory – Sekundärspeicher

* Harddisk / SSD (Langzeitspeicher)
* Über I/O Ports angeschlossen

Bussystem

* Adressbus
  + CPU legt fest, **wohin** geschrieben / gelesen wird.
  + Anzahl Adressen 2^n
* Control Bus
  + CPU legt fest, **ob** gelesen oder geschrieben wird.
  + Kontrollieren den Zeitlichen Ablauf des Zugriffs[[1]](#footnote-1)
* Data Bus
  + **Übertragung** der Daten -> Analogie: Inhalt eines Briefes
    - Schreiben: CPU stellt Daten bereit, Memory übernimmt
    - Lesen: Memory stellt Datenbereit, CPU übernimmt

Übersetzen von Programmen



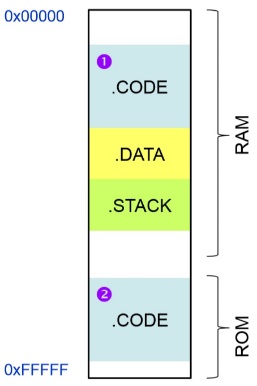
Adressierungsarten

Direkte Adressierung:

MOV AL, DS:[3d] ; Lade Inhalt aus Speicherzelle Nr. 3 nach AL

Indirekte Adressierung:

MOV AL, [BX] ; Inhalt von BX = Speicherzelle, deren Inhalt nach AL

Register Adressierung:

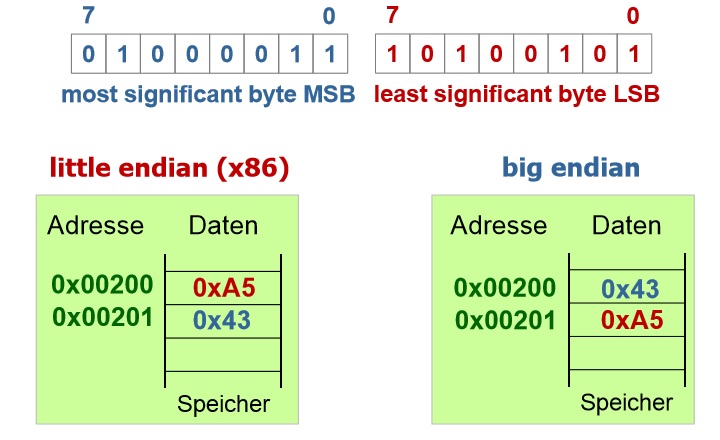
MOV AL, BL

Immediate Adressierung:

MOV AL, 87d

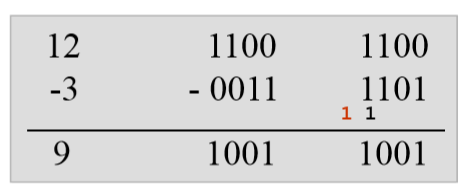
Implizite Adressierung:

PUSH AX ; Speichert Inhalt von AX auf einem Stack



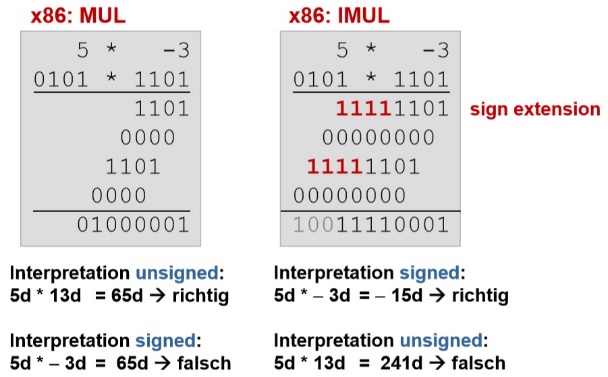
Negative Zahlen (2er Komplement)

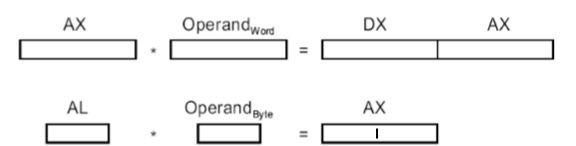
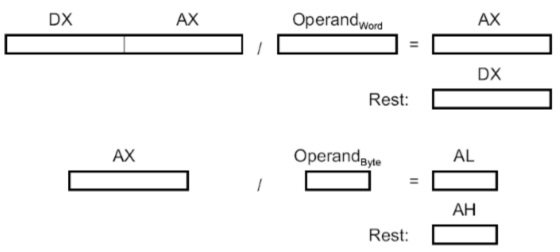
-5d = 5d (0101) negieren -> 1010 -> + 1 = 1011 (-5d)

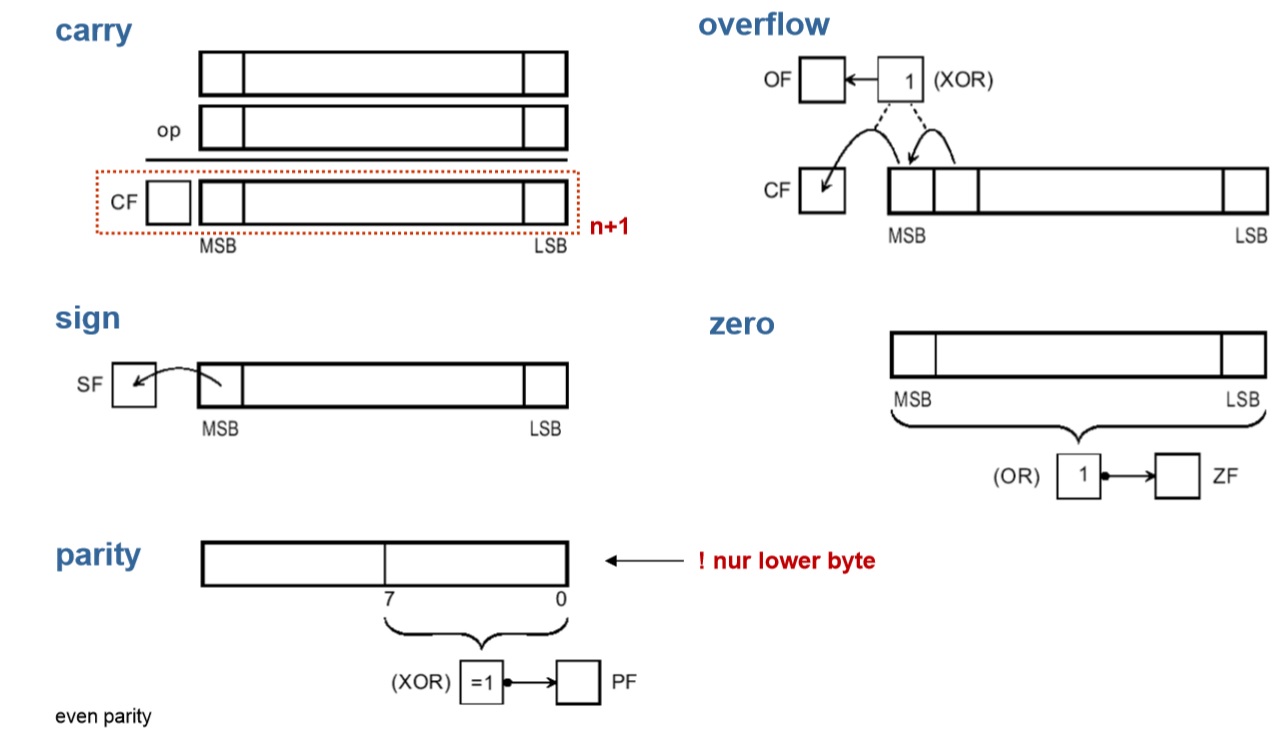
Borrow = Carry invertiert

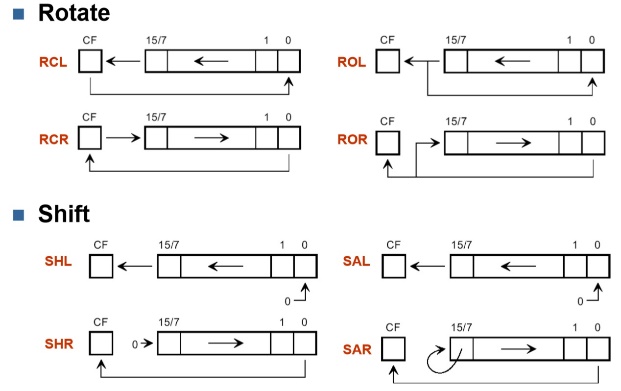
Bsp: Borrow = 1 -> Carry = 0

Multiplikation

Es wird immer mit AX bzw. AL multipliziert.  
Bsp: MUL BX





Logische Instruktionen

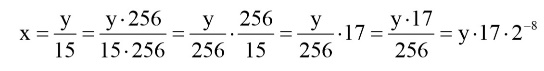
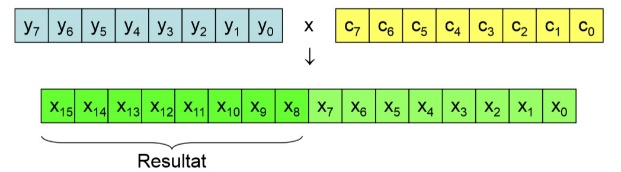
Bits löschen: AND  
Bits setzen: OR  
Bits invertieren: XOR

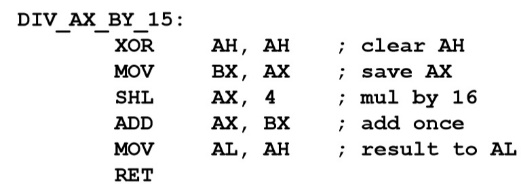
Syntax Shift / Rotate Instruktionen

SHL AH, n  
SHL AH, CL (Nur mit CL möglich)

Division durch Konstante

Division durch Multiplikation ersetzen: Division durch 256 nicht nötig:



Bsp. Division von AL durch 15 (Multiplikation 17)

While Loops

JMP abfrage

do: …

abfrage: CMP \_nr,10  
 JL do

If-then-else Abfrage

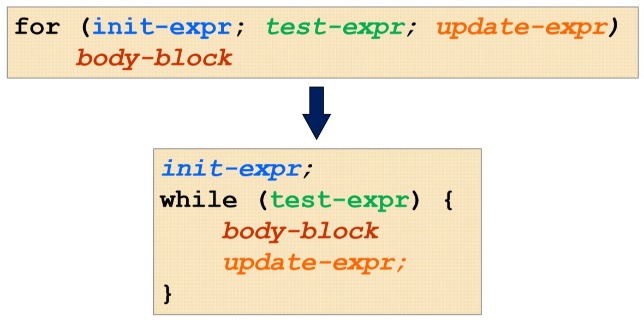
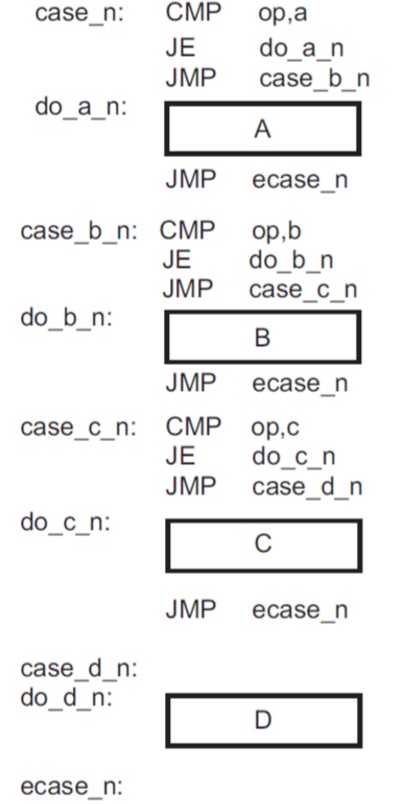
CMP \_nr, 0  
JL true  
JMP false

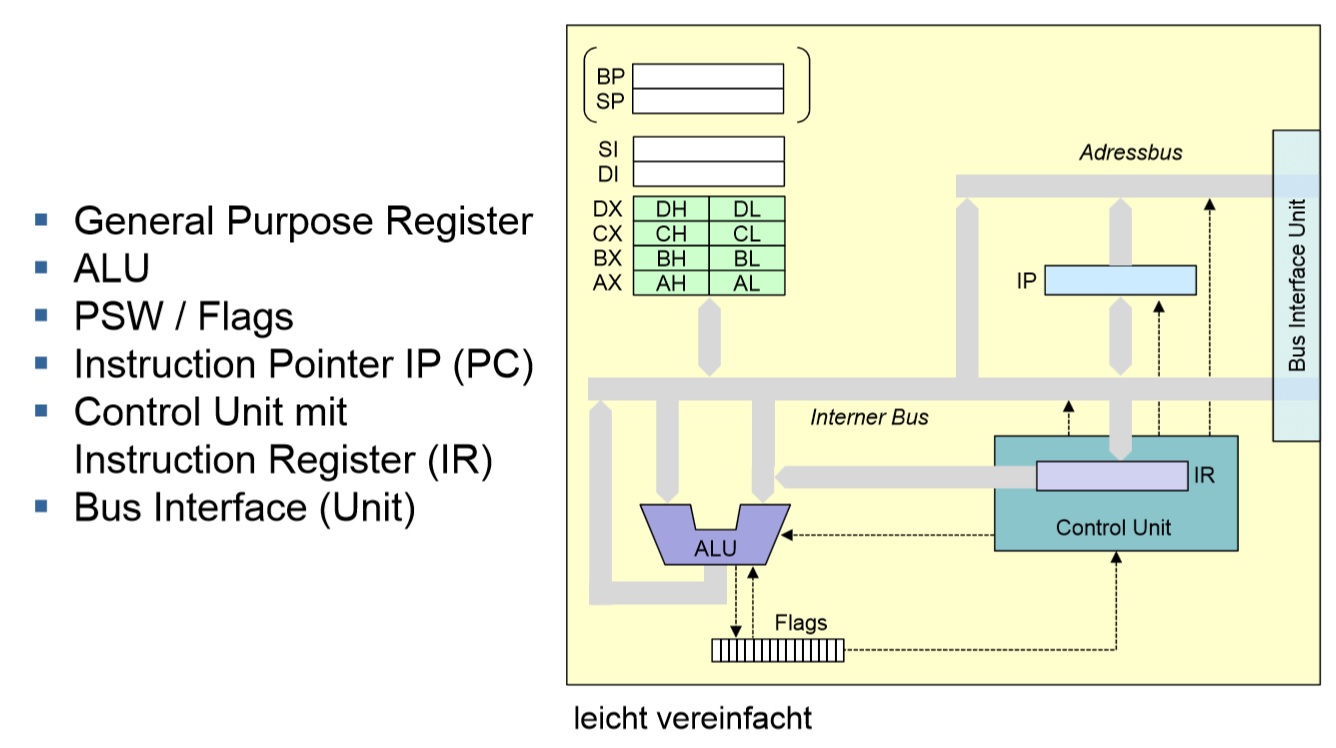
true: ….  
 JMP done

false: …

done: …

For-Loop (Wird in While umgewandelt) Switch Statement





1. Wann sind Adressen und Daten gültig [↑](#footnote-ref-1)