

Übung RA, Kapitel 1.5

1. Beantworten Sie bitte folgende Repetitionsfragen
 1. Beschreiben Sie in eigenen Worten und mit einer Skizze die Schichtung einer Multilevel Maschine.
Folie 5, rechte Seite
 2. Welche Grundlegenden Operationsarten gibt es bei den Maschinenbefehlen?
MOVE Daten herumschreiben (lesen, schreiben)
Programm Ablauf Steuerung (Branch, Jump)
Rechnen (Arithmetisch & Logisch)
 3. Skizzieren Sie einen Drei-Adress Befehl.
ADD adr1, adr2, adr (adr1 + adr2 in adr3 speichern)
 4. Beschreiben Sie mit einer Skizze und ein paar Stichworten, was folgende zwei Adressierungsarten bedeuten: absolute Adressierung und unmittelbare Adressierung
Absolute Adressierung: Wert steht an Adresse: LDA 7 (Lade Akku A mit dem Wert welcher an Adresse 7 steht)
Unmittelbare Adressierung: Wert steht direkt beim Befehl: LDA #7 (Lade Akku A mit dem Wert 7)
 5. Wozu dient der Instruction Pointer? Wie nenn man ihn auch noch?
ProgramCounter, zeigt auf den nächsten auszuführenden Befehl
 6. Wozu kann ein Stack verwendet werden?
Rücksprung Adresse sichern, Parameter Wert übergeben, Zwischenspeicher
 7. Wozu wird ein Cache verwendet?
Schneller Zwischenspeicher

2. Mikrocontroller

Aufgabe 1 Mikrocontroller

Was ist ein Mikrocontroller? (Stichworte)

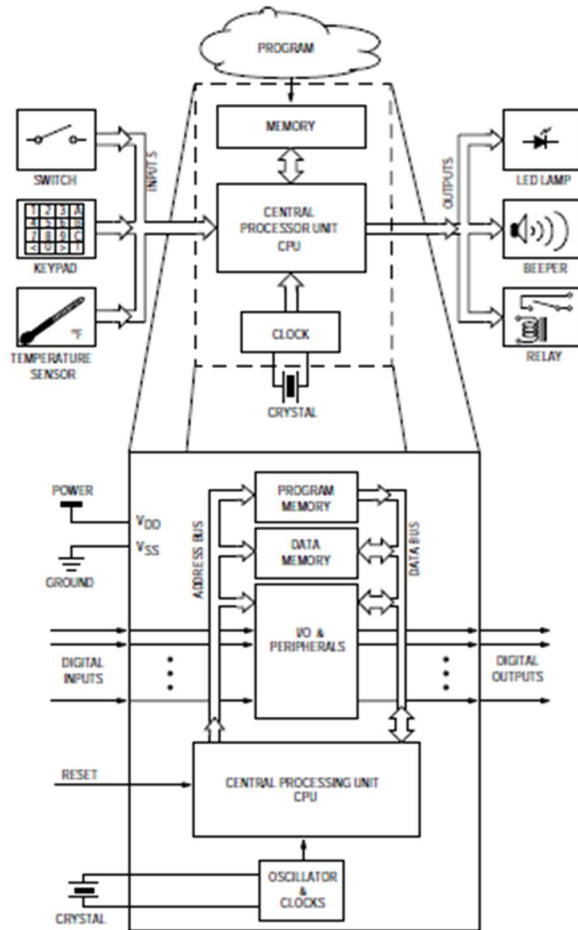


Figure 2. Expanded View of a Microcontroller

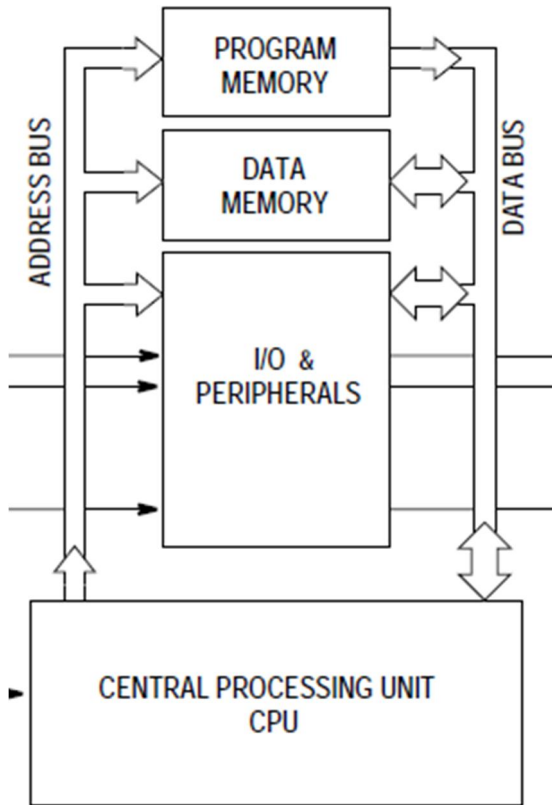
Als Mikrocontroller (auch μ Controller, μ C, MCU) werden Halbleiterchips bezeichnet, die mit dem Prozessor mindestens Peripheriefunktionen auf einem Chip vereinen. In vielen Fällen befindet sich der Arbeits- und Programmspeicher ebenfalls teilweise oder komplett auf dem gleichen Chip. Ein Mikrocontroller ist praktisch ein Ein-Chip-Computersystem. Für manche Mikrocontroller wird auch der Begriff System on a Chip oder SoC verwendet.

Auf modernen Mikrocontrollern finden sich häufig auch komplexe Peripheriefunktionen wie z. B. CAN- (Controller Area Network), LIN- (Local Interconnect Network), USB- (Universal Serial Bus), I²C- (Inter-Integrated Circuit), SPI- (Serial Peripheral Interface), serielle oder Ethernet-Schnittstellen, PWM-Ausgänge, LCD-Controller und -treiber und Analog-Digital-Wandler. Einige Mikrocontroller verfügen auch über programmierbare digitale und/oder analoge Funktionsblöcke.

Quelle: <http://de.wikipedia.org/wiki/Mikrocontroller>

3. Aufgabe 2 Architektur

Um welche Architektur handelt es sich bei diesem Mikrocontroller?



Weil es nur einen Datenbus für Befehl und Daten gibt ist es eine „Von Neumann“ Architektur

4. Aufgabe 3 Mnemonics

(a) Wie lautet die binäre Darstellung der Mnemonischen Codes RTS?

RTS => HEX: 81 (\$81, 81H, 0x81)=> 10000001

(b) Welche Funktion führt dieses "Mnemonic" aus?

RTS = Return from Subroutine => Beendet ein Subroutine

(c) Mit welchen Adressierungsarten kann ADD verwendet werden? Geben Sie dazu zwei Beispiele.

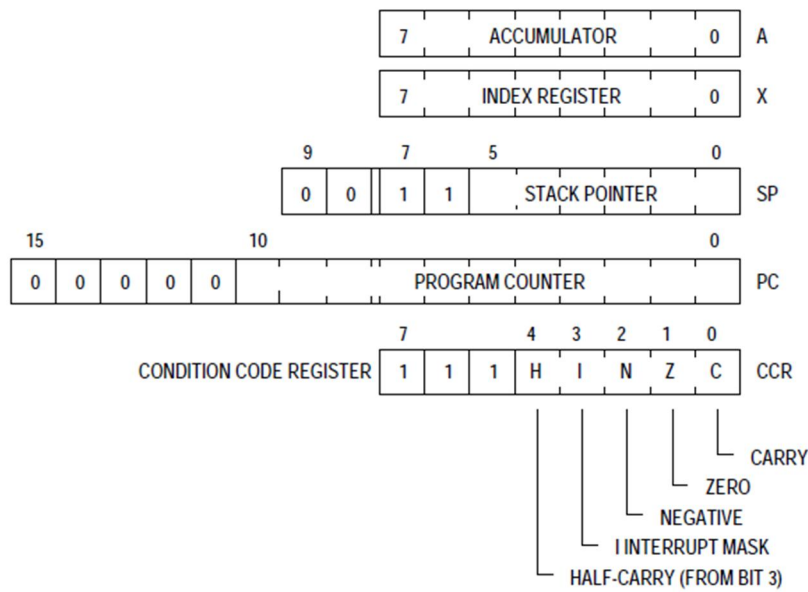
Source Forms,
Addressing
Modes, Machine
Code, and Cycles

Source Forms	Addressing Mode	Machine Code		HCMOS Cycles
		Opcode	Operand(s)	
ADD (opr)	IMM	AB	ii	2
ADD (opr)	DIR	BB	dd	3
ADD (opr)	EXT	CB	hh ll	4
ADD,X	IX	FB		3
ADD (opr),X	IX1	EB	ff	4
ADD (opr),X	IX2	DB	ee ff	5

5. Aufgabe 4 Register

Wie viele Register weist der M68HC05 auf? Benennen Sie alle Register mit Ihrer Abkürzung.

5 Register:



6. Aufgabe 5 Index Register

Bei Aufgabe 4 sollten Sie auch auf ein Register mit dem Namen "Index Register" gestossen sein.

Wozu könnte dieses Register nützlich sein?

Hilfsregister, Schleifenzähler mit mehr als 8 Bit (Akku genügt nicht mehr), Indexieren einer Tabelle

* DLY50 – Subroutine to delay ~50ms

* Save original accumulator value

* but X will always be zero on return

```

0323 B7 C0    DLY50    STA    TEMP1        ;Save accumulator in RAM
0325 A6 41    LDA    #65          ;Do outer loop 65 times
0327 5F      OUTLP    CLRX          ;X used as inner loop count
0328 5A      INNRLP  DECX          ;0-FF, FF-FE,...1-0 256 loops
0329 26 FD    BNE    INNRLP       ;6cyc*256*500ns/cyc = 0.768ms
032B 4A      DECA          ;65-64, 64-63,...1-0
032C 26 F9    BNE    OUTLP        ;1545cyc*65*500ns/cyc=50.212ms
032E B6 C0    LDA    TEMP1        ;Recover saved Accumulator val
0330 81      RTS              ;Return

```

7. Aufgabe 6 Instruction Set

Wie lautet der binäre Code

(a) des Befehls RSP

0x9C

(b) des Befehls LDA? Was fällt Ihnen dabei auf?

0xA6, 0xB6, 0xC6, 0xD6, 0xE6, 0xF6

Verschiedene Adressierungs-Möglichkeiten

Aufgabe 7 **Programmanalyse**

(a) Im folgenden Programmabschnitt wird ein Wert in den Akkumulator A eingelesen. Aus welcher Adresse wird der Wert gelesen? Wie heisst die Adressierungsart?

```
0003          SAM    EQU    $03          SAM equal an 8-bit value
1400          LARRY  EQU    $1400       LARRY equal a 16-bit value
0100          ORG    $100              Set program starting point
0100 ae 02    TOP    LDX    #$02       Initialize index register
0102 a6 05          LDA    #$05       Read value into A
```

0x05 wird in den Akku geladen, unmittelbare Adressierung (immediate). Der Wert wird aus der Adresse 103 gelesen.

(b) Im folgenden Programmabschnitt wird ein Wert in den Akkumulator A eingelesen. Aus welcher Adresse wird der Wert gelesen? Wie heisst die Adressierungsart?

```
0003          SAM    EQU    $03          SAM equal an 8-bit value
1400          LARRY  EQU    $1400       LARRY equal a 16-bit value
0100          ORG    $100              Set program starting point
0100 ae 02    TOP    LDX    #$02       Initialize index register
0102 b6 05          LDA    $05         Read value into A
```

Der Wert aus der Adresse 0x05, absolute / direkte Adressierung

Aufgabe 8 * **Programmanalyse**

Folgendes Programm wird von START bis END abgearbeitet. Welche Zahl steht am Schluss in der Speicherzelle \$00FF?

```
0100 9C          START  RSP          Reset SP to $00FF
0101 cd 02 00          JSR    SUB      Call SUB
0104 cd 02 00          JSR    SUB      Call SUB again
0107 9d          END    NOP          Done
.....
.....
0200 81          SUB    RTS          Just Return
```

Es steht noch immer die letzte Rücksprungadresse im Stack: 0107

Aufgabe 9 * **Subroutinen Verschachtelung**

Wie gross ist die maximale Schachtelungstiefe für Subroutinen? Geben Sie allfällige Randbedingungen genau an.

Bis der StackPointer überläuft. Je nach Init 32 resp. 64 Locations (Adressen)

Stackgrösse ist 64Bit

Jeder Subroutine Aufruf belegt zwei 8-bit Speicherzellen (JSR Beschreibung Seite 261)

Keine Interrupts, weil diese auch Stack brauchen.