

1 Notizen

1.1 03.04

Ein PIC-Microcontroller muss mit der Software programmiert werden. Es muss 3 bis 4 Wochen vor der Klausur abgeschlossen werden. Man hat jedoch bis zum letzten Termin Zeit. Es gibt zusätzlich Punkte für eine frühzeitige Abgabe. Programmiert wird in C, C# oder Java aber anderen gehen auch.

Plan für heute Konzept muss entwickelt werden.

- Interne Struktur festlegen?
- Wie soll die Oberfläche aussehen?
- Wie hab ich es mir vorgestellt was kam raus?

Funktionalität es soll es muss nur angezeigt werden. Im Hintergrund muss nicht genau das ablaufen (keine Binär Addierer sondern die Rechnungen). Alles in Zahlen machen besser als mit Strings.

Wenn der Befehl gefunden wurde.

Wenn man am 15.05 abgibt gibt es 5 Punkte extra (Es gibt zwischen 64 und 80) Punkten

2 Konzept

2.1 Gui

SDL2.0 Gui-Klasse

2.2 Programmiersprache

C Sprachen bzw. C++ (gut für GUI) C++

2.3 Skizze Programm

Orientierung an dem Simulator -> Register anzeigen, Code der durchlaufen wird, usw.

2.4 Einlesen der LST-Datei

Muss noch ein Lösung gefunden werden. Goto, Label usw. Scanner Klasse.

2.5 Funktionen implementieren

Alle Funktionen aus dem Bewertungsbogen implementieren andlw usw. Engine-Klasse.

```
if(befehl && 0x3F00)==0x3000 → MOVLW
if(befehl && 0x3F00)==0x3E00 → ADDLW
if(befehl && 0x3F00)==0x0700 ⇒ ADDWF
if(befehl && 0x3F00)==0X3900 ⇒ ANDLW
MOVLW;
```

```
Ergebnis = Befehl && 0x00FF
W-Reg = Ergebnis
```

```
    ADDLW;
Ergebnis = Befehl && 0x00FF
Ergebnis = Ergebnis + WReg
if Ergebnis == 0 → Z-Flag setzen
else → Z-Flag zurücksetzen
if Ergebnis > 255 → Set Carry
else → Rest
```

```
    ADDWF;
FReg = Befehl && 0x007F
DBit = Befehl && 0x0080
if FReg == 0 ⇒ FReg = FSR (indirekte Addressierung)
Erg = W-Reg + Datenspeicher[FReg]
if D-Bit == 0 → WReg = Erg
else Datenspeicher[FReg] = Erg (FReg ist ein Register welches selbst definiert
werden kann)
```

```
    ANDLW;
```

```
    Temp1 = Befehl && 0x000F
Temp1 = Temp1 + (WReg && 0x0F) if Temp1 > 15 → SetDC
else ⇒ RestDC
```

2.6 Addressmaschinen