

1 Konzept

1.1 Gui

SDL2.0 Gui-Klasse

1.2 Programmiersprache

C Sprachen bzw. C++ (gut für GUI) C++

1.3 Skizze Programm

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

1.4 Einlesen der LST-Datei

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

1.5 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 \longrightarrow Rest

ADDWF;
FReg = Befehl && 0x007F
DBit = Befehl && 0x0080
if FReg == 0 \implies FReg = FSR (indirekte Addressierung)
Erg = W-Reg + Datenspeicher[FReg]
if D-Bit == 0 \longrightarrow 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 \longrightarrow SetDC
else \implies RestDC

1.6 Einlesen der Datei

getline statt scanf

1.7 Datenspeicher

Ein Array der Größe 1024 als Variablentyp Integer Array[1024][2] High und Low

1.8 Addressmaschinen

2 Notizen

2.1 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 Addierter 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

3 Aufgaben

3.1 Funktion des Codes

- Alle Befehlen müssen funktionieren und den Datenspeicher verändern
- Direkte und indirekte Addressierung
- Timerfunktion ohne / mit Berücksichtigung der Bits im OPTION-Register (e / o)
- Interrupt für Timer 0
- Interrupt für INT (RB0)
- Interrupt RB4-RB7
- ADDWF PCL mit Berücksichtigung von PCLATH (TPicSim101)(theor. Für CALL und GOTO; geprüft an Hand des Codes)

3.2 Funktionen der Gui

- Datenspeicher visualisieren
- Den Text der LST datei visualisieren
- Die RA und Tris Pins müssen visualisiert werden und per Maus gesetzt werden können
- Breakpoints also Debugging mit der Visualisierung der LST dateien
- Highlighting des aktuellen (nächsten) Befehls im LST-Fenster
- Spezialregister darstellen
- Laufzeit darstellen
- Steuerpult mit Start Stop und Step in

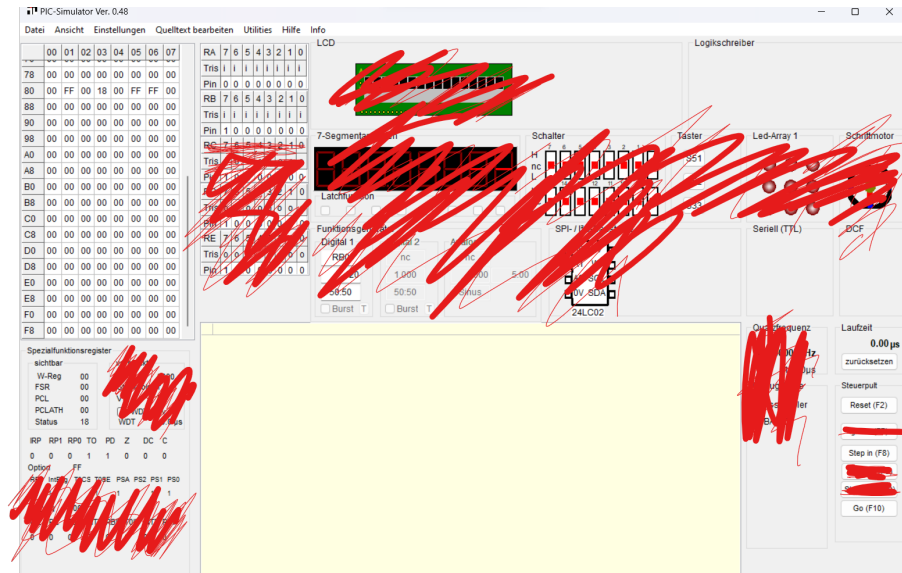


Abbildung 1: lehmann-Pic bearbeitet

Datenregister		Spezialfunktionenregister				Steuerpult	
RA	76543210	W-Reg	00	IRP	0	Z	0
TRIS	11111111	FSR	00	RPI	0	PC	0
DIN	00000000	DCH	00	RPO	0	C	0
RB	76543210	RELATH	00	TD	01		
BR	11111111	Status	18	PD	1		
PR	00000000						
0		L S T Date				Reset	
0						Step in	
0						Go	
0						Stop	
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							
0							