Praktijkexamen computerhardware 13/01/2021 17u30-20u00

Documentatie

De datasheet en labonota's kan je terugvinden in het zip-bestand dat je van indianio.ugent.be hebt gedownload en dat je hebt uitgepakt in de map C:\temp of C:\VMSharedDrive.

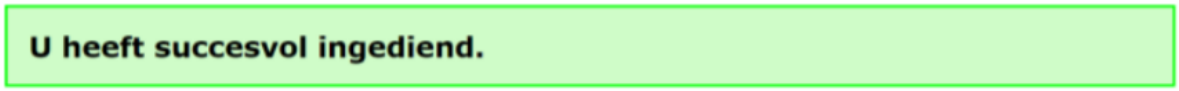
Richtlijnen

De oplossingen van de opgaven schrijf je weg in de bestanden 1.asm, 2.asm, 3.asm, 4.asm en 5.asm die je zonet hebt uitgepakt in de map C:\temp of C:\VMSHaredDrive.

Voorzie je programma's van voldoende commentaar en zorg dat ze assembleerbaar zijn.

Bij het beëindigen van het examen, ten laatste om 20u00 maak je een zip-bestand met daarin de vijf asm-bestanden die je indient op indianio.ugent.be.

Opgelet, wanneer je na het uploaden van het zip-bestand onderstaande boodschap in het browservenster niet ziet verschijnen, is het uploaden mislukt!



Enkel de ingediende bestanden (1.asm, 2.asm, 3.asm, 4.asm en 5.asm) zullen bij quotering worden geraadpleegd!

Verder laat je het toestel gewoon aanstaan.(niet uitloggen, geen shutdown of reboot uitvoeren)

Enkele nuttige pagina's in de datasheet:

- Overzicht interrupt vectoren (pagina 147)
- Overzicht Special Function Registers (pagina 138 onderaan)
- Overzicht verschillende instructies (pagina 121)

Opgaven

1. (5 pt) Gegeven het volgende 80x86-codefragment:

```
cwd
neg AX
adc DX, DX
```

Wat is de betekenis van het bovenstaande codefragment indien je weet dat AX=input en DX=output en dat de instructie **neg AX** wordt uitgevoerd als:

```
if (AX ==0) {
    carry=0;
} else {
    carry=1;
}
AX= -AX; (AX wordt dus vervangen door zijn tweecomplement)
```

De instructie cwd (convert word to double word) zal het tekenbit van de accumulator AX dupliceren over het volledige register DX (cfr. sign extension). Wanneer het getal in AX negatief is, zal DX de waarde 0xffff bevatten. Wanneer het een positief getal bevat, is dit 0x0000.

Ter info, het tekenbit is het meest beduidende bit van het register. Een getal in 2-complementsnotatie voldoet aan de voorwaarde $-x = \bar{x} + 1$. Om dus bv. de voorstelling van -7 te bepalen, moet je het complement nemen van het getal 7 en er 1 bij optellen.

Tot slot is de werking van de instructie adc gelijk aan de werking van de 8051-instructie addc (add with carry).

Vul in het bestand 1.asm drie gehele getalwaarden in. M.a.w. wat zit er in DX wanneer het getal dat zich in AX bevindt, positief, negatief of gelijk is aan 0. Schrijf de waarde van DX in decimale voorstelling en als geheel getal!

2. (5 pt) Gegeven het volgende 80x86-codefragment:

```
xor AX, BX
xor BX, AX
xor AX, BX
```

Schrijf in het bestand **2.asm** in één tekstregel wat de functie is van bovenstaand codefragment.

3. (5 pt) Schrijf in **3.asm** een 8051-assembleertaalcodefragment (enkel de nodige instructies en geen volledig programma dus) dat op een zo efficiënt mogelijke manier het getal dat zich in de accumulator bevindt deelt door 2.

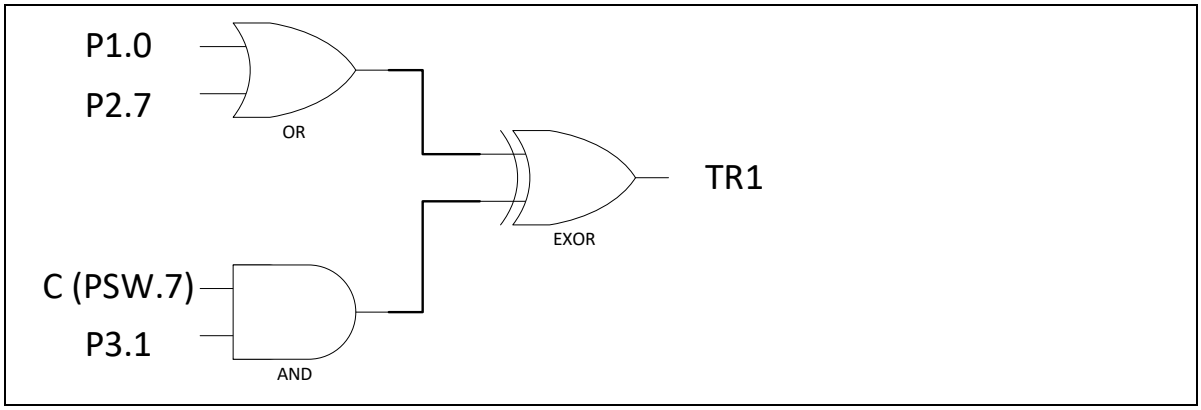
4. (20 pt) Gegeven volgende C-code:

```
void strcpy(char *dest, char* src){
    if (*src!=0){
            *dest=*src;
             src++;
             dest++;
             strcpy(dest,src);
    }
}
void main(void) {
    char src [5]={'a','b','c','d',0};
    char dst [5];
     strcpy(dst,src);
    while(1){ };
}
```

In het hoofdprogramma heeft de array src als startadres 30H en de array dst 50H. De ASCII-waarde van de letter 'a' is 61H en voor de overige letters gewoon oplopend. 'b' heeft dus als ASCII-waarde 62H, 'c' 63H, enz. .

Zet bovenstaande C-code letterlijk (zonder iets aan te passen of iets te optimaliseren) om naar assembleertaal en schrijf dit volledige programma weg in het bestand **4.asm**. Zorg dat er bij het uitvoeren van de subroutine **strcpy** geen enkel register permanent gewijzigd wordt.

5. (25 pt) Gegeven onderstaand schema waarvan de functie totaal onbelangrijk is:



Schrijf een volledig 8051-assembleertaalprogramma dat continue deze logische schakeling doorloopt (veronderstel gerust dat er geen andere poortpinnen gebruikt worden). Wanneer TR1 op 1 gezet wordt, wordt de timer uiteraard gestart en wordt er op poortpin P0.5 een blokgolf van 10 KHz uitgestuurd. De schakeling wordt ondertussen verder doorlopen. Veronderstel dat de klokfrequentie van de CPU 25 MHz bedraagt en veronderstel ook dat de frequentie voor timer 1 nog eens intern gedeeld wordt door 12.

De werkingsmode van timer 1 doet er niet toe. Werk dus gerust met een 16-bit timer of met een 8-bit timer al naargelang je eigen voorkeur.

De code zelf schrijf je weg in **5.asm**!