**Test Computerhardware labo 2016 - 2017 (eerste zit)**

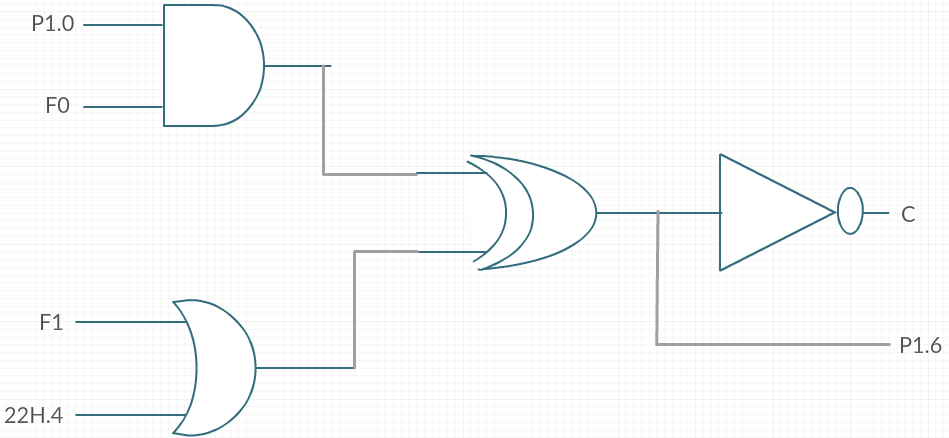
Wat ter beschikking gesteld was:

* De labocursus in PDF-formaat (“cursus\_labo\_hardware.pfd”)
* De datasheet van het microcontroller (“c8051f12x.pdf”)
* De Windows calculator
* De testopgaven (zowel in PDF-formaat, als afgeprint)

De test was opgedeeld in 5 opgaven, waarbij elke opgave in een aparte, reeds aangemaakte .asm bestand moest opgelost worden

**1) Opgave 1:**

Maak een programma die de volgende logische circuit nabootst:



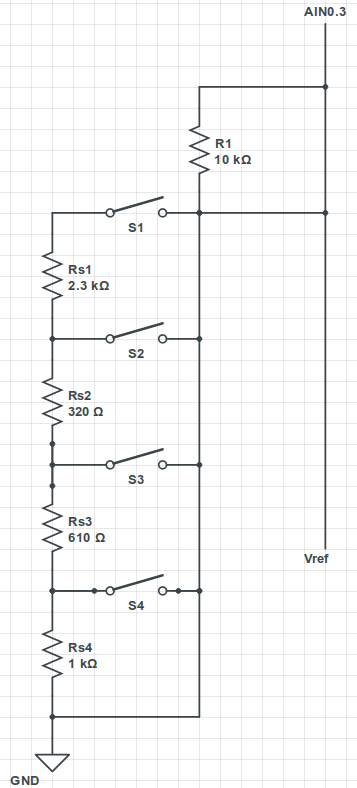
Dit heeft een continue werking dus je mag niets van de oorspronkelijke gegevens wijzigen.

**2) Opgave 2:**

Bij deze opgave wordt een knoppenschakeling van 4 knoppen (of was het schakelaars) d.m.v. de pin AIN0.3 analoog verbonden met de microcontroller. Ook was aan Poort 1 van de microcontroller een LED-bar verbonden. Je moest een programma maken waarmee bij het indrukken van S1 de LED op P0.0 zou aangaan, bij het drukken van S2 de LED op P0.1, bij het drukken van S3 de LED’ P0.0 en P0.1 en tenslotte bij het drukken van S4 de LED op P0.2. (zie schakeling hieronder).

Bovendien moest bij het aankomen van een signaal op INT1 het programma volledig gestopt worden.  
In de papieren versie van de test werd ook eerst in een tabel gevraagd om de voltages bij elke schakelaar te berekenen en te schrijven en tevens eronder de 12-bit digitale waarde van deze voltages (Vin).

(de getekende circuit kan fout zijn, dus gelieve het te verbeteren of het me laten weten)



Je moest werken met ADC0 (dacht ik).

**3) Opgave 3:**

Als ik het me nog herinner, moest je programma schrijven om een schuifregister na te bootsen. Je MOEST werken met timer 3, en je moest om de 0,7 ms een bit van R0 naar buiten brengen, eerst LSB en daarna MSB. Je moest verplicht werken met interrupts.

meer weet ik ivm deze opgave niet meer

4) Opgave 4:

vermenigvuldig de accumulator met 61,25 zonder mul/div/en MAC0. Het resultaat komt in B (MSB) en A (LSB) terecht

5) Opgave 5:

Geef de assembly versie van de volgende C-programma, de stack moet gebruikt worden.

void bereken\_lengte(char \* c\_string, int &lengte) {

if(c\_string[0] != 0) {

++lengte;

bereken\_lengte(++c\_string, lengte);

}

}

int main() {

char \* c\_string [10]= “hallo”;

int lengte = 0;

bereken\_lengte(c\_string, lengte);

while(1);

}

Gebruik maken van de stack was verplicht. In de opgave stond vermeld dat “c\_string” begon aan het adres 40H en de variabele “lengte” aan het adres 30H. Ook kreeg je als extra informatie dat de ASCII-getalwaarde van “a” gelijk was aan 61H en dat de ints uit acht bits bestonden.

Je moest ervoor zorgen dat aan het einde van het hoofdprogramma alle registers hun oorspronkelijke waarde terugkregen.