MIC Aflevering 3 – Division 16bit med 8 bit

Det første der sker i min kode er at portene B,D sættes op til output og input. På port D sættes der pull-up for at det kan virke som input.

Derefter sættes stacken og variablerne op.

Derefter tømmes registrene som benyttes for at være sikker på at der ikke ligger noget eller at de register som normalt plejer på ZH/ZL ikke har noget liggende.

MAIN LOOP

Herefter starter MAIN loopet. Det første der sker er at loade de to 16-bit tal ind som skal ligges sammen. Herefter kaldes ”SUM16” som ligger de to 16-bit tal sammen.
Herefter vil resultatet af additionen ligge i ”NumLSB/NumLSB” og derfor skrives der nu det næste tal der skal skrives ind i ”SecondNumLSB/SecondNumMSB” hvorefter der igen kaldes ”SUM16”.

Dette gentages indtil alle tallene er lagt sammen.
Inden der divideres skrives 5 ind i registeret ”Denominator” så den er klar til divisionen.
Der kaldes så ”DIV16_8” subrutinen for at dividerer. Når det er sket så komplimenteres registrene inden der hoppes til PRINT_DIODE for at vise den korrekte værdi da det er aktive lave.

Her efter hoppes så til PRINT_DIODE.

SUM16

I Sum16 ligges 2 16-bits tal sammen ved først at adderer de to tals LSB og herefter adderer med carry de to tals MSB. Her efter returnerer den til hvor den blev kaldt fra.

DIV16_8

I DIV16_8 divideres et 16-bit tal med et 8-bits tal.
Det første der sker er at Kvotienten inkrementeres med 1 vha. ADIW. Herefter trækkes der 5 fra NumLSB. Derefter trækkes der 0 med carry fra NumMSB.
Dette virker ved at tallet der divideres med, trækkes først fra den lave del af tælleren. Hvis der ikke er nok til det så sættes carry flaget. Så når du så trækker 0 fra tællerens MSB så vil den trække carry fra også.
Derefter testes så som carry flaget fra MSB er sat. Hvis det ikke er sat er der altså mere at trække fra med og der jumpes til DIV16_8 for at gentage.
Hvis carry derimod er sat så har vi trukket fra en gang for meget og derfor breakes der ikke. Der fortættes i stedet og ligges 5 til tælleren. Da vi fandt ud af at vi havde trukket fra en gang mere end den kunne og derfor ligger vi de 5 der blev trukket fra til tælleren igen.

Derefter bliver der så af samme grund trukket en fra kvotienten. Der bliver jo ligesom gået 1 skridt tilbage.

Det sidste der sker er at resten vi har i NumLSB bliver flyttet til Denominator og kvotienten bliver flyttet til NumLSB/NumMSB. Da det er et krav i opgaven.

Herefter returneres hvorfra der blev kaldt.

PRINT_DIODE
I PRINT_DIODE tjekkes hvilke knapper trykkes og viser derefter den korrekte værdi.

Det første der sker er at den tjekker med Sbis om ”PD:6” er sat. Hvis den er sat betyder det at knappen S10 ikke er trykket ned. Da knapperne er aktiv lave. Hvis knappen ikke er trykket ned vil den skippe den næste instruktion og derfor ikke hoppe til S10.

Det samme gøres så for PD:2 mhs. til S11 knappen.
Hvis begge knapper ikke er trykket ned så forsættes der til ”Snone”. Her skrives 0xFF til R19 som så sendes til displayet, hvilket slukker displayet.

S10
Hvis der jumpes til S10 så tjekkes der først om den anden pind er trykket ned. Da hvis den er det skal der jumpes til Sboth da begge knapper er trykket ned.
Hvis ikke begge knapper er trykket ned så skrives NumLSB til displayet og der jumpes til PRINT_DIODE.

S11
Her tjekkes ligesom før om den anden knap er trykket ned. Hvis det ikke er tilfældet skrives NumMSB. til displayet. Og jumper derfra til PRINT_DIODE

Sboth
Denne køres kun hvis begge knapper er trykket ned og den sender ”resten” som ligger i denominatoren til displayet.
Der jumpes så til PRINT_DIODE.

Frederik Mazur Andersen