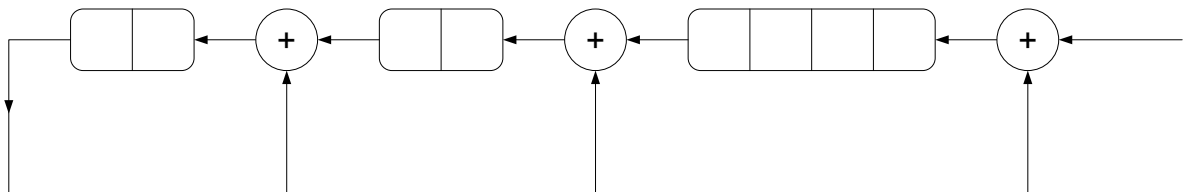


Oefeningenreeks 8: Subroutines en het gebruik van een digitale temperatuursensor

1. Schrijf een eerste subroutine voor het bepalen van het product van twee getallen. De twee getallen worden voor het aanroepen van de subroutine op de stapel geplaatst. Bij deze oefening mag je gebruikmaken van de MUL-instructie. De uitvoer van de subroutine bevindt zich in de accu(LSB) en in het B-register(MSB).
2. Schrijf een recursieve subroutine die de faculteit berekent van het getal dat zich in de accu bevindt. Het resultaat komt terecht in het B-register.
3. Herschrijf de subroutine uit vraag 1 maar zonder de MUL-instructie te gebruiken en zonder gebruik te maken van MAC0. Zoek eerst een gepast algoritme om twee 8-bit getallen te vermenigvuldigen. De uitvoer moet zich opnieuw in de accumulator(LSB) en het B-register(MSB) bevinden.
4. Onderstaand schema toont een hardware-implementatie van een 8-bit CRC-berekening (cyclic redundancy check). Zoals te zien is, wordt er gebruikgemaakt van één 8-bit register en drie XOR-poorten. Telkens als er aan de linkerzijde een bit binnenschuift, worden alle bits in het 8-bit register één positie naar links opgeschoven. Op discrete plaatsen wordt er een XOR-bewerking toegepast tussen het door te schuiven bit en het bit dat uit het register vrijkomt.



Bijvoorbeeld, bij de volgende data 11001100110 wordt het volgende berekend:

Inhoud CRC-register

In te voegen
bit

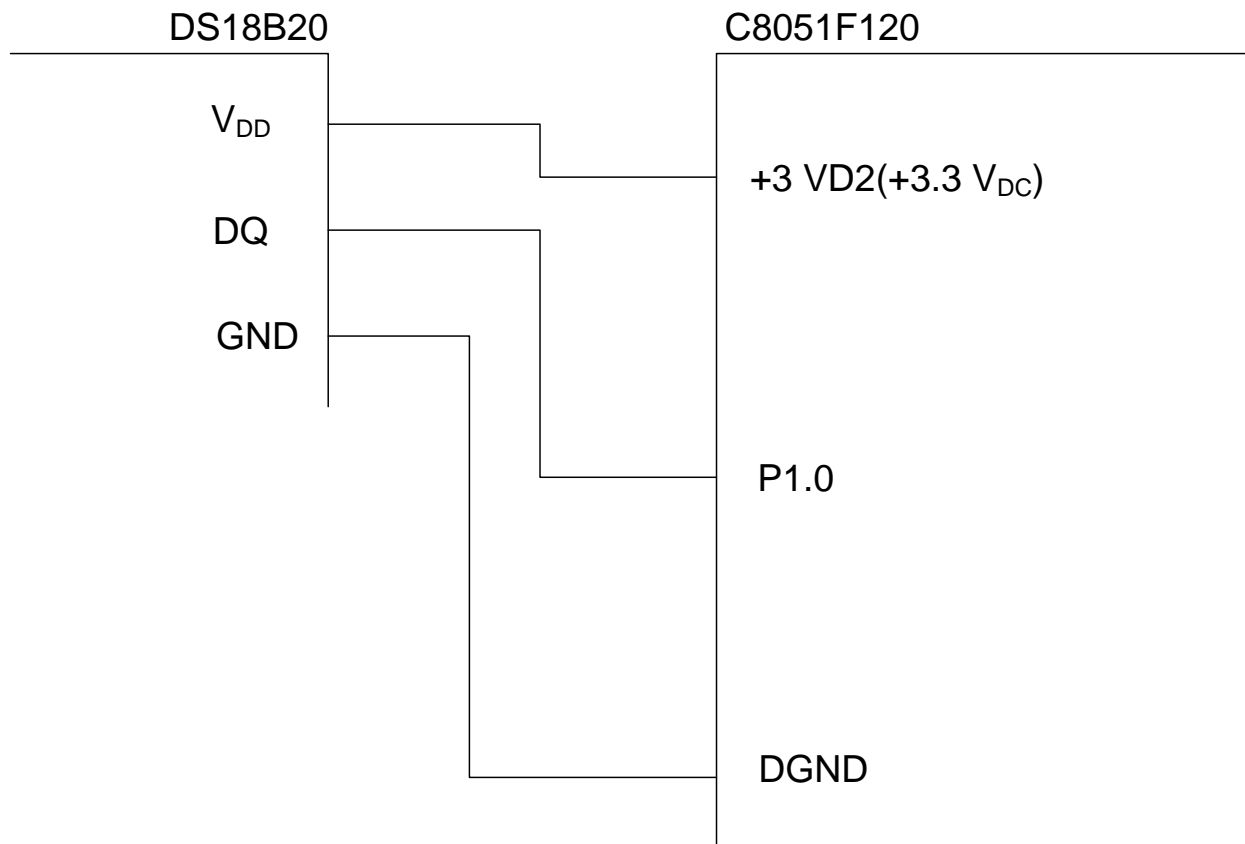
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1	0
0	0	0	0	0	1	1	0	0
0	0	0	0	1	1	0	0	1

0	0	0	1	1	0	0	1	1
0	0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	0	0
1	1	0	0	1	1	0	0	1
1	<u>1</u>	0	<u>0</u>	1	0	0	<u>0</u>	1
1	<u>1</u>	0	<u>0</u>	0	0	0	<u>0</u>	0
1	<u>1</u>	0	<u>1</u>	0	0	0	<u>1</u>	STOP

De CRC bedraagt dus 11010001, dit is de inhoud nadat alle databits doorgeen het register werden geroteerd. Bemerk dat de begintoestand overeenstemt met de waarde 0.

Schrijf een volledig ASM-programma dat alle bits van alle geheugenvakjes gelegen in het gesloten interval [20H, 30H] de CRC-8 berekent volgens het gegeven schema. De inhoud van ieder geheugenvakje wordt bit per bit in het CRC-register geschoven, te beginnen bij het minstsignificante bit. Er wordt dus begonnen bij het minstsignificante bit van 20H en gestopt bij het meestsignificante bit van adres 30H.

5. Schrijf een ASM-programma dat de temperatuur, opgeslagen in het geheugen van de DS18B20, opvraagt. Ga in debug-mode na of de gemeten waarde met de werkelijkheid overeenstemt.



Een digitale sensor is eigenlijk opgebouwd uit een analoge sensor, een ADC en een kleine hoeveelheid geheugen. De eindgebruiker hoeft dus niets te weten over het type sensor, de analoog-naar-digitaal omzetting en de eventuele niet-lineariteit van de sensor. De gebruiker moet daarentegen wel weten hoe de opgemeten temperatuur uit het geheugen van de digitale sensor moet worden gehaald.

Bij deze oefening wordt gebruikt gemaakt van een DS18B20 digitale temperatuursensor. Deze sensor kan temperaturen meten in een bereik van -55°C tot 125°C. De populariteit van deze sensor ligt wellicht aan het feit dat de voeding van de sensor parasitair kan gebeuren. Dit betekent dat de sensor wordt gevoed via de datalijn zelf i.p.v. via een externe bron. Aangezien er bij parasitaire voedingen een condensator wordt gebruikt om de voeding van de sensor constant te houden, wordt er verwacht dat de datalijn op diverse momenten verplicht op een logische 1 moet worden gebracht. Aangezien dit de complexiteit van de gegevensoverdracht vergroot wordt bij deze oefening gebruikgemaakt van een externe bron.

Bestudeer eerst de datasheet van de DS18B20 en bekijk hoe de temperatuurgegevens kunnen worden opgevraagd. De communicatie met de DS18B20 verloopt over één digitale I/O-lijn. Het protocol dat master en slave met elkaar moeten spreken is beschreven in de bovenvernoemde datasheet.