

WEEK 2

ONDERWERPEN

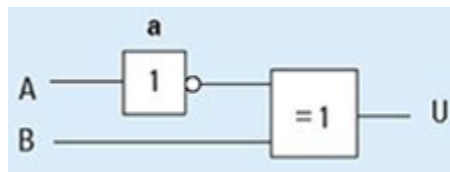
- typen computers
- 8 great ideas
- organisatie van de computer
- CPU intern, instructies uitvoeren
- geheugen systeem
- adres- en databus
- pipelining
- byte ordering
- de AVR MCU

OPDRACHT 1 : SCHAKELINGEN MET POORTEN

- a) Geef de waarheidstabellen voor onderstaande schakelingen a t/m d.
 b) Kun je voor schakeling d een eenvoudiger oplossing bedenken?

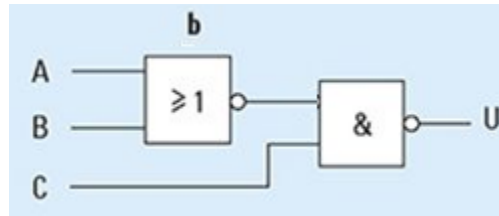
a) (= Dijkstra 2.1c zie p 366)

$$U = !A \text{ XOR } B = \text{XOR}(!A, B) = 1001$$



A	B	!A	U
0	0	1	1
0	1	1	0
1	0	0	0
1	1	0	1

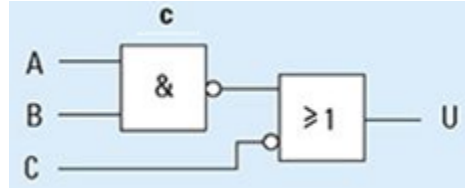
b) (is NIET 2.1d) $U = ! (A + B) * C$



A	B	C	A+B	U1=! (A+B)	U=! (U1*C)
0	0	0	0	1	1
0	0	1	0	1	0
0	1	0	1	0	1
0	1	1	1	0	1
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	1	0	1

c) (= Dijkstra 2.1e zie p 366) $U = \text{NAND}(A,B)+C' = 1 \times 0 \text{ rest } 1 = \text{NAND met 3 ingangen}$

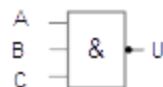
$$U = ! (A * B) + !C$$



A	B	C	A*B	U1=! (A*B)	!C	U= U1+!C
0	0	0	0	1	1	1
0	0	1	0	1	0	1
0	1	0	0	1	1	1
0	1	1	0	1	0	1
1	0	0	0	1	1	1
1	0	1	0	1	0	1
1	1	0	1	0	1	1
1	1	1	1	0	0	0

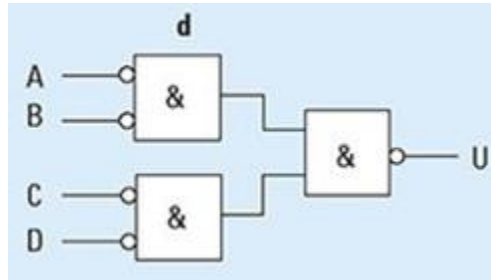
Eenvoudiger oplossing

$U = \text{NAND}(A,B)+C' = 1 \times 0 \text{ rest } 1 = \text{NAND met 3 ingangen}$



= Dijkstra 2.1f zie p 366 = $AND(A', B', C', D') = 1 \times 0 \text{ rest } 1 = OR \text{ met } 4 \text{ ingangen}$

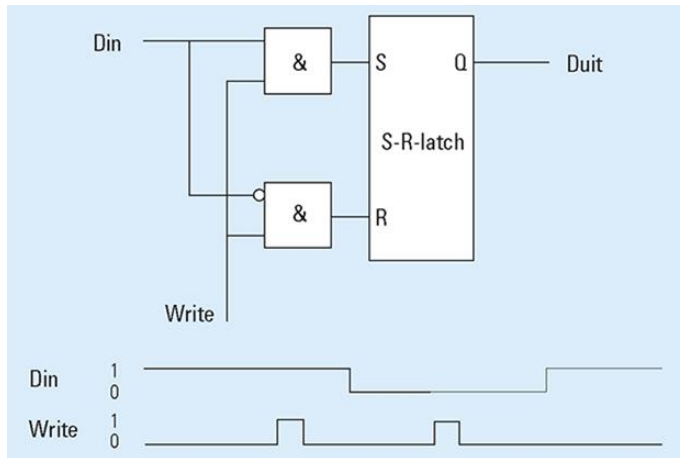
$$U = ! (A * B) * (C * D)$$



A	B	C	D	!A	!B	U1=(!A*!B)	!C	!D	U2=(!C*!D)	U = !(U1*U2)
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	0	0	1
0	0	1	0	1	1	1	0	1	0	1
0	0	1	1	1	1	1	0	0	0	1
0	1	0	0	1	0	0	1	1	1	1
0	1	0	1	1	0	0	1	0	0	1
0	1	1	0	1	0	0	0	1	0	1
0	1	1	1	1	0	0	0	0	0	1
1	0	0	0	0	1	0	1	1	1	1
1	0	0	1	0	1	0	1	0	0	1
1	0	1	0	0	1	0	0	1	0	1
1	0	1	1	0	1	0	0	0	0	1
1	1	0	0	0	0	0	1	1	1	1
1	1	0	1	0	0	0	1	0	0	1
1	1	1	0	0	0	0	0	1	0	1
1	1	1	1	0	0	0	0	0	0	1

OPDRACHT 2 : SCHAKELINGEN MET POORTEN

Gegeven onderstaande schakling. (De getekende waarden voor Din en Write zijn slechts voorbeelden).



Dit is 2.24 uit Dijkstra.

a) Maak de volgende tabel af :

W	Din	S	R	Q
1	1	1	0	1
1	0	0	1	0
0	1	0	0	hold
0	0	0	0	hold

b) Formuleer in één zin hoe de uitgang Duit zich gedraagt als functie van beide ingangen.

Conclusie : als Write=1 dan wordt Din "ingeklokt". Write werkt dus als een "strobe-sigitaal", d.w.z. alleen als Write even hoog is wordt de ingang ingeklokt.

M.a.w. het is een flip-flop.

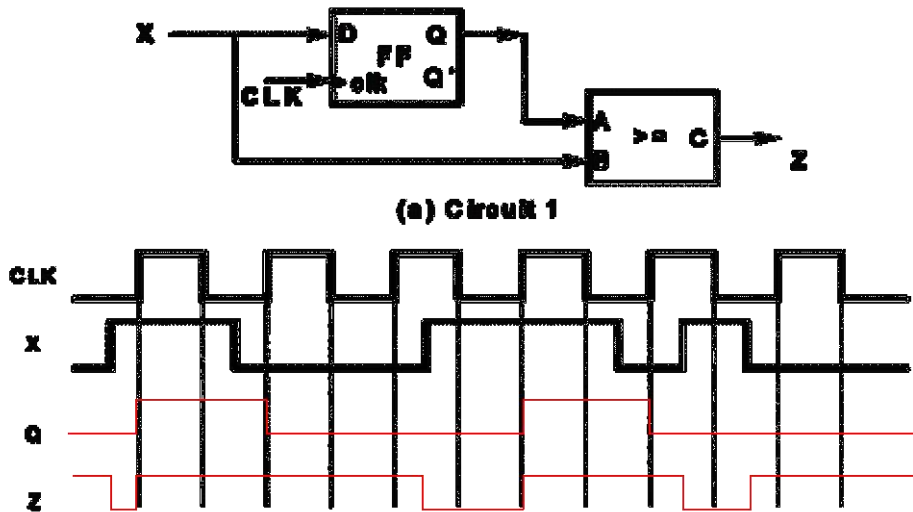
OPDRACHT 2 : SCHAKELINGEN MET POORTEN

In onderstaande schakeling is FF een **flip-flop**. Dit betekent : als de klok-ingang CLK van 0 naar 1 gaat dan wordt Q gelijk aan D. De vergelijker heeft als output een "1" als $A \geq B$, anders "0". (CLK staat voor "clock").

Teken het timing diagram met daarin ingevuld het verloop van Q en Z.

Antwoord

NB. Belangrijk is dat Q alleen kan veranderen als CLK van 0→1 gaat.



OPDRACHT 4: IPV4-ADRES EN SUBNETMASKER (VOORUITBLIK OP THEMA 2.2 NSE/SE NETWERKEN)

IPv4 is de hedendaagse standaard als het gaat om netwerken. In de wereld van IPv4 wordt een 32 bits-adres gebruikt om een systeem (device/host) te adresseren. Een voorbeeld van zo'n adres is 145.37.148.110. Een andere belangrijk begrip in IPv4 is het subnetmasker. Een voorbeeld is 255.255.255.0 (ook wel bekend als /24).

Gegeven de volgende situatie:

IPv4-adres 1, device A: 145.37.148.110

IPv4-adres 2, device B: 145.37.148.138

Subnetmasker: 255.255.255.128

4a) Iemand stelt jou de vraag In hoeverre deze adressen behoren bij hetzelfde subnetwerk? Bepaal dit aan de hand van het volgende recept:

- Schrijf beide adressen volledig binair uit
- Schrijf het netwerkmasker volledig binair uit
- Doe een AND van adres 1 met het netwerkmasker. Doe hetzelfde voor adres 2.
- ALS (en ALLEEN ALS) beide uitkomsten gelijk zijn DAN behoren de beide adressen bij hetzelfde netwerk

4b) Stel X = aantal aangesloten 1-en achter elkaar (v.l.n.r.) in de binaire notatie van het subnetmasker. / X is dan een andere notatie van het subnetmasker. Wat is de waarde van X ?

adres 1	145	37	148	110
binair	10010001	00100101	10010100	01101110
masker	255	255	255	128
binair	11111111	11111111	11111111	10000000
adres 1 AND masker	10000001	00000101	10000100	00000000
adres 2	145	37	148	138
binair	10010001	00100101	10010100	10001010
masker	255	255	255	128
binair	11111111	11111111	11111111	10000000
adres 2 AND masker	10000001	00000101	10000100	10000000

OPDRACHT 5: OPERATING SYSTEMS, MMU EN BITS (VOORUITBLIK OP THEMA2.2 NSE/SE OPERATING SYSTEMS)

Een cruciale taak van een modern *Operating System* is het beheren van het geheugen van een systeem. Een kerntaak hierbij is het alloceren (toewijzen) van geheugen aan processen EN zorgen dat processen onderling niet bij elkaars geheugen kunnen. Het OS maakt hiervoor gebruik van een hardware-component (die meestal in de CPU zit) die *Memory Management Unit* heet.

Moderne MMU's maken gebruik van een techniek die paginering heet (Engels: *paging*). Hiervoor wordt het geheugen ingedeeld in vakken (pagina's/*pages*) van een bepaalde omvang. Deze pagina's liggen achterelkaar in het geheugen, lineair doorgenummerd. Stel, we hebben een pagina-grootte van 1KByte. Op welke pagina, en op welke offset bevindt zich dan geheugenadres 2069.

Bepaal dit op 2 manieren:

- Bepaal hoe vaak 1Kbyte past in 2069. Noteer dit en schrijf ook de rest op.
- Schrijf 2069 binair op. "Hak" deze uitkomst doormidden, waarbij rechts 10 bits staan. Wat zijn de waarden van het linkergedeelte resp. het rechtergedeelte?
- Controleer of de uitkomsten van a en b gelijk zijn. (tip: dit zou wel zo moeten zijn)

paginagrootte	1024	
adres	2069	
aantal keren		
passend	2	
rest	21	
adres binair(16bits)	0000100000010101	
rechterstuk (10 bits)	0000010101	21
linkerstuk (2+ bits)	000010	2

OPDRACHT 6: SYMMETRISCHE ENCRYPTIE (VOORUITBLIK OP THEMA 2.3 NSE)

Een populaire, moderne, encryptietechniek is AES. Dit wordt onder andere gebruikt bij Wifi (WPA2 – PreSharedKey).

Een volledig uiteenzetting van AES gaat hier te ver. Maar, een van de cruciale stappen die tijdens de versleuteling(encryptie) wordt gedaan wordt is een XOR. Een XOR van de (tussen de partijen) afgesproken GEHEIME sleutel en de data die verstuurd moet worden. In de praktijk wordt hiervoor een sleutel gebruikt die 128 tot 256 bits lang is. In deze opdracht hanteren we voor het gemak een 32 bits sleutel.

Stel het te versleutelen bericht is 'SCMI'. De geheime sleutel (ook bekend als *presharedkey* of *passphrase*) is 'DEEP'.

a) Codeer het bericht met de sleutel.

Zoek de bijbehorende ASCII-waarden op van het bericht en de sleutel. Schrijf deze daarna binair uit. Doe een XOR van bericht en de sleutel.

b) Decodeer het versleutelde bericht

Neem de uitkomst van a) en doe opnieuw een XOR met de sleutel.

c) Controleer of de uitkomst van b) gelijk is aan het originele bericht.

bericht	S	C	M	I
ascii	83	67	77	73
binair	01010011	01000011	01001101	01001001
sleutel	D	E	E	P
ascii	68	69	69	80
binair	01000100	01000101	01000101	01010000
XOR	00010111	00000110	00001000	00011001
sleutel binair	01000100	01000101	01000101	01010000
XOR 2	01010011	01000011	01001101	01001001
	S	C	M	I

OPDRACHT 7 : DATABUS EN ADRESBUS

- Een adresbus bestaat uit 6 lijnen, hoeveel adressen/geheugenplaatsen kan je hiermee bereiken ?
 2^6
- Een geheugenkaart bevat 32K RAM geheugenadressen, hoeveel adreslijnen zijn nodig om dit geheugen te adresseren ?
 ${}^2\log(32k) = {}^2\log(2^5 * 2^{10}) = 15$
- Een geheugenkaart heeft een databus van 16 bit en een adresbus van 32 bit. Hoeveel bytes bevat het geheugen ?
Meerder antwoorden zijn mogelijk, maar je mag dan aannemen dat een register uit het geheugen dan ook 16 bit is. Dus $2 * 2^{32}$ byte = 8Gbyte

OPDRACHT 8 : DE MEMORYMAP

Een systeem beschikt over een databus van 16 bit en een adresbus van 32 bit.

- Hoeveel geheugenlocaties zijn er ? 2^{32}
- Wat is het hoogste adres uitgedrukt als hexadecimaal getal ? 32 enen binair, dus hex 0xFFFF
- Uit hoeveel bytes zal een geheugenlocatie (waarschijnlijk) bestaan ? databus is 16 bit = 2 bytes, dus logisch zou zijn $2^{32} * 2 \text{ bytes} = 2^{30} * 8 = 8 \text{ GB}$

OPDRACHT 9 : EEN VIDEOFRAME

Gegeven een kleuren display waarbij elke kleur (R,G,B) wordt gecodeerd met 8 bit voor elke pixel. Het videoframe heeft een grootte van 1280×1024 pixels.

- Hoeveel geheugen is nodig om een frame op te slaan ?
 $1280 * 1024 * 3 \text{ bytes}$
- Hoe lang zal het minimaal duren om een frame te versturen over een 100 Mbit/s netwerk link ?
 $1280 * 1024 * 3 * 8 / 100 * 10^6 = 315 \text{ us}$

OPDRACHT 10 : CISC OF RISC

In de datasheet van de ATmega32 kun je op de eerste bladzijde zien dat de CPU een RISC architectuur heeft. Stel dat je dit niet zou weten en je alleen de instructieset van de ATmega32 zou hebben. Noem tenminste 3 eigenschappen van de instructieset waaraan je kunt zien dat het een RISC CPU betreft.

- als je de instructies bekijkt, zie je dat ze allemaal eenvoudig zijn
- bijna alle instructies duren 1 klokcyclus
- er zijn veel algemene registers (r0..r31)
- er is een vaste instructielengte (bijna alle instructie 16 bit)
- er is een beperkte set addressing modes
- dataoverdracht geheugen gaat alleen via LOAD en STORE instructies, alle andere operaties werken op interne registers

OPDRACHT 11 : DE SYSTEEMKLOK

De systeemklok heeft een frequentie van 2 MHz. De prescaler heeft als waarde 2000. We willen elke 0,4 seconde een interrupt genereren. Hoeveel tellerstanden moet de counter doorlopen voordat een interrupt wordt gegenereerd ?

$\text{één periode teller} = 2000 / 2.000.000 = 1/1000 \text{ seconde} = 1\text{ms}$
 $\text{aantal tellerstanden} = \text{duur} / \text{periode} = 0.4 / 0.001 = 0.4 * 1000 = 400$

OPDRACHT 12 : CPU-KLOK EN CPI

Gegeven zijn drie verschillende CPU's P1, P2, en P3 met alle drie dezelfde instructieset. F staat voor frequentie en CPI staat voor het aantal klok-cycles per instructie. Verder is gegeven :

- a) Wanneer P1 t/m P3 een programma van 10 seconden uitvoeren, hoeveel CPU-cycles kost dit en hoeveel instructies worden er dan uitgevoerd ?

Antwoord

Aantal cycles in 10 sec = $10 * F$

aantal instructies in 10 sec = $10 * F / \text{CPI}$

	F CPU in GHz	#cycles in 10 sec *miljard	CPI	#instr in 10 sec *miljard
P1	3	30	1.5	20
P2	2,5	25	1.0	25
P3	4	40	2.2	18

- b) Welke CPU heeft de hoogste performance in termen van instructies per seconde ? **P2**

OPDRACHT 13: DE NAAM ATMEGA32

In de datasheet van de ATmega32 staat dat de PC 14 bits breed is (zie hoofdstuk "AVR ATmega32 Memories", paragraaf over Flash Program Memory).

- a) Uit hoeveel geheugenplaatsen bestaat het code segment ?

PC register is 14 bit (zie datasheet "The ATmega32 Program Counter (PC) is 14 bits wide, thus addressing the 16K program memory locations."

- b) Wat is (dus) het grootste adres op de ATmega32 (hexadecimaal genoteerd) ?
 $2^{14} - 1$, oftewel $0x4000 - 1 = 0x3FFF$ (zie ook de memory map in datasheet)

- c) Uit hoeveel bits bestaat één geheugenlocatie van het code segment ?

1 adres = 1 instructie = 16 bit

- d) Hoe groot is het geheugensegment dus in kbytes ?

Er zijn dus $2^{14} = 16\text{k}$ geheugenplaatsen van elk 16 bit breed = totaal 32 k byte. Vandaar de naam ATmega32.

OPDRACHT 14: BYTE ORDER

Laat in een figuur zien hoe de waarde 0xABCDEF12 in het geheugen wordt geplaatst bij een little-endian en een big-endian machine, waarbij gegeven is dat een geheugenlocatie 16-bit breed is.

Little-Endian		Big-Endian	
Address	Data	Address	Data
12	ab	12	12
8	cd	8	ef
4	ef	4	cd
0	12	0	ab