

Practicum 4: Digitale Techniek - Combinatorische logica en geheugenelementen

Semih Can Karakoç (695258)

5 juni 2023

Inhoudsopgave

1	Inleiding	4
2	Probleem- en doelstelling	5
3	Practicumbenodigdheden	6
4	Theoretisch kader	7
4.1	De basispoorten	7
4.2	Waarheidstabellen	7
4.3	Wetten van de Booleaanse algebra	8
4.4	7-segmentdisplay	9
4.5	Flip-flop's	10
4.6	Het 3910 digiboard: versie 2	11
5	Methode en aanpak	13
6	Combinatorische logische functie: MUX	15
6.1	Stel de waarheidstabel op voor de 4-bit multiplexer	16
6.2	Herleid vanuit de opgestelde waarheidstabel de bijbehorende logische uitdrukking	16
6.3	Teken de logische schakeling die deze uitdrukking representeert	16
6.4	Bouw de schakeling op het digiboard	17
6.5	Controleer de werking van de logische schakeling	17
7	Combinatorische logische functie: BCD-encoder	18
7.1	Stel een BCD-tabel op voor decimale getallen 0 t/m 9	19
7.2	Herleid vanuit deze opgestelde tabel de bijbehorende logische uitdrukking	19
7.3	Teken de logische schakeling die deze uitdrukking representeert	20
7.4	Bouw de schakeling op het digiboard	21
7.5	Controleer de werking van de logische schakeling door de BCD-uitgangen aan te sluiten op de BIN/7SEG decoder	21
8	Teller (2N-toestanden)	22
8.1	Ontwerp met behulp van 4 J-K-flip-flops een teller met 2N toestanden	23
8.2	Stel aan de hand van het ontwerp, de bijbehorende toestandstabel op	23
8.3	Hoe wordt deze teller ook wel genoemd	24
8.4	Bouw deze schakeling op het digiboard	25
8.5	Sluit de uitgang van elke flip-flop aan op een LED	25

8.6	Sluit op de klokingang van de J-K-flip-flops een blokgolf met een frequentie van 1 Hz aan	26
8.7	Controleer of de schakeling alle toestanden doorloopt	26
8.8	Eigenschappen en tijdvolgordediagram	27
9	Teller (N-toestanden)	28
9.1	Ontwerp met behulp van 4 J-K-flip-flops een teller met N-toestanden	29
9.2	Stel aan de hand van het ontwerp, de bijbehorende toestandstabel op	29
9.3	Hoe wordt deze teller ook wel genoemd	30
9.4	Bouw deze schakeling op het digiboard	31
9.5	Sluit de uitgang van elke flip-flop aan op een LED	31
9.6	Sluit op de klokingang van de J-K-flip-flops een blokgolf met een frequentie van 1 Hz aan	32
9.7	Wat gebeurt er	32
9.8	Hoe zorgen we ervoor dat de teller start	32
9.9	Controleer of de schakeling alle toestanden doorloopt	32
9.10	Eigenschappen en tijdvolgordediagram	33
10	Conclusie	34
11	Bronvermelding	37

1 Inleiding

Tijdens dit practicum is er geëxperimenteerd en onderzocht naar de werking van de multiplexer, BCD-encoder en ringtellers. Het doel van dit practicum is dan ook om de aspecten van deze logische functies te begrijpen en in de praktijk toe te kunnen passen. Om de bovenstaande practicumdoel te behalen zijn er in totaal vier experimenten uitgevoerd. Daarbij zijn de bijbehorende deelopdrachten uitgewerkt in dit verslag.

Dit verslag gaat in op de volgende onderwerpen:

1. Logische schakelingen lezen, bouwen en ontwerpen;
2. Binary Coded Decimal (BCD);
3. 4-bit multiplexer;
4. D-flip-flops en J-K-flip-flops;
5. Ringtellers;
6. Waarheidstabellen;
7. Booleaanse algebra.

NOTE: *alle getekende logische schakelingen in dit verslag zijn zelf gemaakt met draw.io!*

2 Probleem- en doelstelling

Dit verslag bevat de volgende probleemstelling:

- *“Inzicht krijgen in de werking van een aantal complexe logische schakelingen en daarna deze op een efficiënte wijze te ontwerpen en te bouwen?”*

De volgende doelstellingen behoren tot dit practicum:

1. Inzicht te verkrijgen in logische functies die ten grondslag liggen aan de digitale techniek;
2. Waarnemingen te verrichten aan de bijbehorende eenvoudige logische schakelingen;
3. De bijbehorende meetgegevens en waarheidstabellen op een juiste wijze te verkrijgen en te interpreteren;
4. De vergaarde gegevens te verwerken tot een verslag.

Aan het einde van dit verslag onder het kopje 10: *“Conclusie”* zal de gestelde probleemstelling beantwoord worden.

3 Practicumbenodigdheden

Hieronder zijn de benodigdheden op een rijtje gezet die nodig waren voor het uitvoeren van het practicum:

1. Digiboard: het 3910 digiboard is een bord waarop je gemakkelijk de logische schakelingen kan bouwen door gebruik te maken van de ingebouwde poorten. Deze hebben we gebruikt tijdens het practicum. [1];
2. Verbindingssnoeren: deze snoeren zijn nodig om verbinding te maken tussen allerlei elektronische componenten;
3. Draw.io: dit is een online website waarmee je gemakkelijk logische schakelingen kunt ontwerpen;

4 Theoretisch kader

4.1 De basispoorten

In de digitale techniek zijn er drie belangrijke basispoorten, namelijk:

1. De AND-poort: de AND-poort voert een vermenigvuldiging uit tussen de inputs. Dit betekent dat de AND-poort alleen '1' terug geeft wanneer de alle inputs ook '1' zijn, want een vermenigvuldiging met '0' geeft altijd '0' terug. Dit wordt uitgedrukt als: $A \cdot B$;
2. De OR-poort: de OR-poort voert een optelling uit tussen de inputs. Dit betekent dat de OR-poort alleen '1' terug geeft wanneer minimaal een van de twee inputs '1' is, want een optelling met '1' en '0' is altijd '1'. Dit kan worden uitgedrukt als: $A + B$;
3. De NOT-poort: de NOT-poort voert een inversie uit op de input. Dit betekent dat de NOT-poort altijd het tegenovergestelde van de input terug zal geven. Een voorbeeld is: A wordt \overline{A} (A niet) [2].

4.2 Waarheidstabellen

Tijdens het practicum wordt ook gebruik gemaakt van waarheidstabellen. Een waarheidstabel is een tabel die overzicht geeft over alle mogelijke combinaties van inputwaarden en de bijbehorende uitvoerwaarden [3]. De waarheidstabel begint aan de linkerkant met de inputwaarden. Nadat alle inputwaarden zijn genoteerd komen er twee verticale strepen. Deze twee strepen geven aan dat de daarop volgende variabelen uitvoerwaarden zijn [4].

Zie hieronder een voorbeeld van een waarheidstabel die twee inputwaarden heeft: 'A' en 'B', en één uitvoerwaarde heeft, namelijk: ' $A \cdot B$ ' (AND-operatie). Met deze waarheidstabel kun je alle mogelijkheden overzichtelijk terug zien.

A	B	$A \cdot B$
1	1	1
1	0	0
0	1	0
0	0	0

4.3 Wetten van de Booleaanse algebra

In de Booleaanse algebra zijn er een aantal belangrijke wetten die je kunt toepassen om Booleaanse expressies te kunnen vereenvoudigen, namelijk:

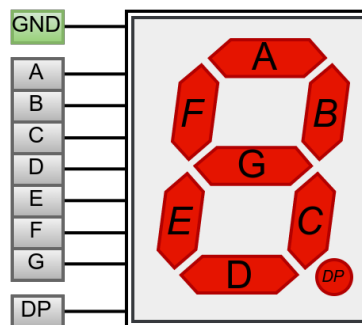
- Associativiteit: $A + (B + C) = (A + B) + C$ en $A \cdot (B \cdot C) = (A \cdot B) \cdot C$;
- Commutativiteit: $A + B = B + A$ en $A \cdot B = B \cdot A$;
- Absorptie: $A + (A \cdot B) = A$ en $A \cdot (A + B) = A$;
- Complement: $A + \bar{A} = 1$ en $A \cdot \bar{A} = 0$;
- Distributiviteit: $A + (B \cdot C) = (A + B) \cdot (A + C)$ en $A \cdot (B + C) = (A \cdot B) + (A \cdot C)$;
- Idempotentie: $A + A = A$ en $A \cdot A = A$;
- De Morgan: $\overline{A \cdot B} = \bar{A} + \bar{B}$ en $\overline{A + B} = \bar{A} \cdot \bar{B}$;
- Redundantie: $A \cdot B + A \cdot \bar{B} = A$ en $A \cdot \bar{B} + B = A + B$;
- Dubbele negatie: $\overline{\bar{A}} = A$ [5].

4.4 7-segmentdisplay

Een 7-segmentdisplay is een display dat bestaat uit zeven lijnvormige segmenten die samen een 8 vormen. De segmenten zijn gelabeld van ‘a’ tot en met ‘g’, zie afbeelding 1. Door deze segmenten op een unieke manier aan en uit te zetten, is het mogelijk om elk getal van 0 tot en met 9 te representeren op het display [6].

- Het getal 0 is te representeren met segmenten: a, b, c, d, e, f ;
- Het getal 1 is te representeren met segmenten: $b \ \& \ c$;
- Het getal 2 is te representeren met segmenten: a, b, g, e, d ;
- Het getal 3 is te representeren met segmenten: a, b, g, c, d ;
- Het getal 4 is te representeren met segmenten: b, c, f, g ;
- Het getal 5 is te representeren met segmenten: a, c, d, f, g ;
- Het getal 6 is te representeren met segmenten: a, c, d, e, f, g ;
- Het getal 7 is te representeren met segmenten: a, b, c ;
- Het getal 8 is te representeren met segmenten: a, b, c, d, e, f, g ;
- Het getal 9 is te representeren met segmenten: a, b, c, d, f .

Daarnaast is het ook mogelijk om hexadecimale getallen te weergeven op het display. Dit kan namelijk met de standaard 0 tot en met F (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F).



Figuur 1: Schema van het 7-segmentdisplay met een “dp” (*decimal point*) ingang.

4.5 Flip-flop's

Een flip-flop is een geheugenelement dat in de elektronica wordt toegepast. Deze "flip-flop" wordt ook wel een "bistabiele multivibrator" genoemd. Elke soort flip-flop bestaat uit een bepaalde soort latch met één klokingang. Het input signaal voor de latch zal alleen doorgevoerd worden zodra de flank van de klokingang actief HOOG staat [7]. In dit verslag wordt er gewerkt met twee soorten flip-flop's, namelijk:

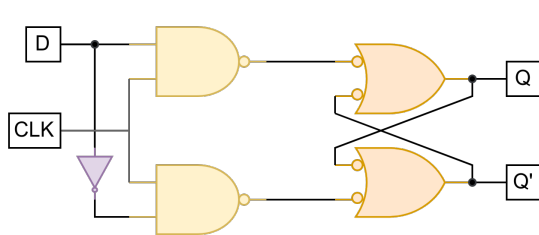
- De D-flip-flop, wordt ook wel "data-flip-flop" genoemd, is een flip-flop dat een data-ingang D en een klokingang heeft;
- De J-K-flip-flop, is vernoemd naar "Jack-Kilby-flip-flop". Het is een flip-flop dat een klokingang en twee data-ingangen J en K bevat.

Zie hieronder de waarheidstabellen van de D-flip-flop en de J-K-flip-flop:

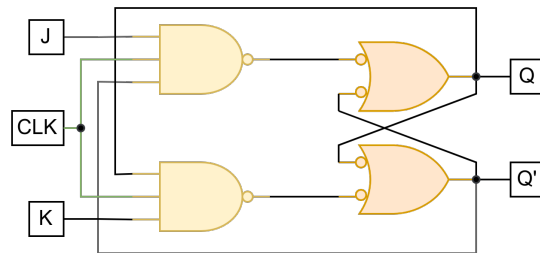
D	Q	\bar{Q}	<i>Comments</i>
1	1	0	<i>SET</i>
0	0	1	<i>RESET</i>

J	K	Q	\bar{Q}	<i>Comments</i>
0	0	Q_0	\bar{Q}_0	<i>No change</i>
0	1	0	1	<i>RESET</i>
1	0	1	0	<i>SET</i>
1	1	\bar{Q}_0	Q_0	<i>TOGGLE</i>

Flip-flop klok signalen kunnen een positieve of een negatieve flank (edge) hebben. Dit houdt in dat de klokpuls pas wordt doorgevoerd aan het begin (positive edge) of einde (negative edge) van een puls. Positive edge klokpulsen worden aangegeven met een pijl omhoog en negative edge klokpulsen worden aangegeven met een pijl omlaag. Daarnaast kunnen flip-flop's worden toegepast in veel applicaties zoals in schuifregisters, frequentiedelers, dataopslag, tellers en nog veel meer.



Figuur 2: Logische schakeling van de D-flip-flop.



Figuur 3: Logische schakeling van de J-K-flip-flop.

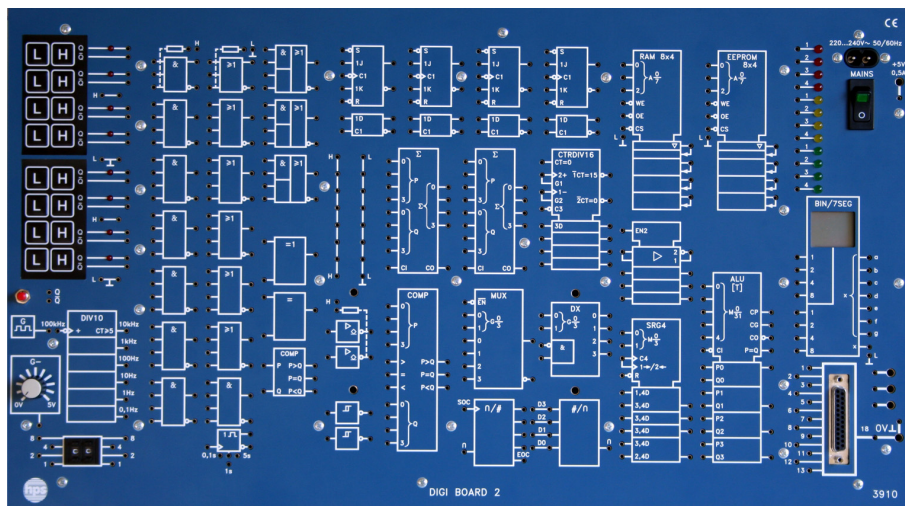
4.6 Het 3910 digiboard: versie 2

De “Type 3910 DIGI BOARD 2” is gefabriceerd door *hps SystemTechnik* en is bedoeld als een universele trainingsbord om studenten te laten leren omgaan met logische schakelingen en poorten in de digitale techniek [8]. Dit digiboard bevat de volgende ingebouwde componenten: [1]

1. 2 input keyboards with 4 pairs of keys (L/H) each;
2. Clock generator with divider, TTL level, crystal-controlled;
3. DC signal source 0...5 V/ 10 mA;
4. Hexadecimal/dual coding switch (double);
5. LED display, divided into 3 groups with the colors red, yellow, green;
6. HIGH/LOW, for tapping HIGH, LOW states;
7. 7-segment display (2-digit), with decoder: dual/7-segment;
8. Adapter (2 mm jacks/SUB-D socket), for adapting 2 mm jacks to SUB-D connector (25-pin), pins 1...13 and 18 assigned;
9. 8 AND gates, with pull-up Resistors, one gate is disconnectable;
10. 6 OR gates, with pull-down Resistors, one gate is disconnectable;
11. 3 AND/OR combi-gates;
12. 1-bit comparator;
13. 4-bit comparator;
14. 4 J-K-flipflops, can also be used as RS flipflops;
15. 4 D-flipflops;
16. 2 adders (4-bit), with input and output carry;
17. Monoflop, settable times: 0.1s; 1s; 5s;
18. Multiplexer, 4 channels;
19. Demultiplexer, 4 channels;
20. Shift register (4-bit), parallel and serial operation possible, bidirectional;
21. ALU, for conducting 16 arithmetic and 16 logical computing operations with 4-bit dual numbers;
22. Binary counter (4-bit), up/down counter;

23. 2 inverters with open collector (pull-up resistors can be connected);
24. 2 Schmitt triggers, inverting;
25. Units complements for negating a 4-bit binary number;
26. Antivalence and equivalence gates;
27. RAM 8x4, static RAM, 8 addresses, 4 bits data width;
28. EEPROM 8x4, storage time without power supply approx. 1 hour;
29. AD / DA converter (4-bit);
30. Two slots for expanding a circuit with additional plug-in modules.

Zie de afbeelding hieronder voor de representatie van het digiboard:



Figuur 4: Foto van de "Type 3910 DIGI BOARD 2".

5 Methode en aanpak

Om dit practicum op een juiste manier uit te voeren is de volgende methodiek toegepast:

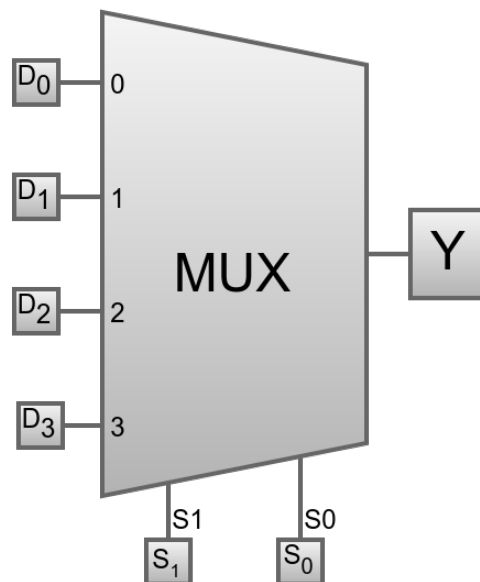
1. De voorbereiding voor het practicum:
 - De practicum benodigdhedenlijst (zie: H3) bekijken en de benodigdheden verzamelen en klaarzetten voor gebruik;
 - Het theoretische kader (zie: H4) van dit verslag even goed doornemen zodat je weet hoe bepaalde materialen en de theorie in elkaar zit;
 - De vragen van het practicum alvast bekijken en vervolgens zoeken naar mogelijke antwoorden zodat je dat (mogelijk) in ieder geval al klaar hebt staan.
2. Het uitvoeren van het practicum:
 - Ga rustig door de vragen heen en noteer je antwoorden, als je iets niet weet kun je het altijd nog op internet zoeken of bij de bijbehorende docent een vraag stellen;
 - Maak foto's van de logische schakelingen die je hebt gebouwd tijdens het practicum. Dit zodat je later in je mogelijke verslag kunt bewijzen dat je er ook echt was en alles werkend hebt gekregen;
 - Ruim alle geleende spullen van school, zoals het digiboard, IC-board en snoeren weer netjes terug in het lab. Bewaar je antwoorden en gemaakte foto's op een juiste manier, zodat je later netjes een verslag kunt schrijven.
3. Het schrijven van het verslag:
 - Maak een document voor dit practicum en geef een gepaste naam aan je document;
 - Zorg ervoor dat je in ieder geval een: voorblad, inhoudsopgave, inleiding, probleemstelling, theoretisch kader, methode, resultaten, conclusie en bronvermeldingen in het verslag hebt staan;
 - Op het voorblad zet je de naam van het verslag, je eigen naam (met evt. studentnummer erbij) en datum;
 - In je inleiding schrijf je kort een introductie over het practicum zelf en geef je het algemene doel van het verslag aan;
 - In het theoretisch kader geef je uitleg over de toegepaste theorie tijdens het practicum, hierbij zijn bronnen erg belangrijk;

- In de methode geef je uitleg over hoe je alles hebt aangepakt en uitgewerkt hebt, zodat de lezer het practicum mogelijk zelf zou kunnen uitvoeren;
- De resultaten/antwoorden splits je op in secties net zoals de vragen zijn gesteld in het practicum bestand. Deze resultaten werk je netjes uit en zet je de gemaakte foto's en schema's in de bijbehorende vraag;
- In de conclusie geef je antwoord op je probleemstelling en gestelde doelen aan de hand van je resultaten en bespreek ook de belangrijke bevindingen die je had tijdens het practicum;
- Tenslotte moet je ook bronvermeldingen hebben. Hierin staan alle gebruikte bronnen, zoals datasheets, artikelen, verslagen, literatuur en overige online bronnen. Dit zodat de lezer kan checken waar je bepaalde informatie vandaan hebt gehaald en mogelijk meer kan lezen over de gebruikte informatie.

6 Combinatorische logische functie: MUX

In de eerste opdracht zijn de volgende deelvragen onderzocht en beantwoord over de multiplexer:

1. Stel de waarheidstabel op voor een 4-bit multiplexer;
2. Herleid vanuit de opgestelde waarheidstabel de bijbehorende logische uitdrukking;
3. Teken de logische schakeling die deze uitdrukking representeert;
4. Bouw de schakeling op het digiboard;
5. Controleer de werking van de logische schakeling.



Figuur 5: Een 4-bit multiplexer (afgekort MUX).

6.1 Stel de waarheidstabel op voor de 4-bit multiplexer

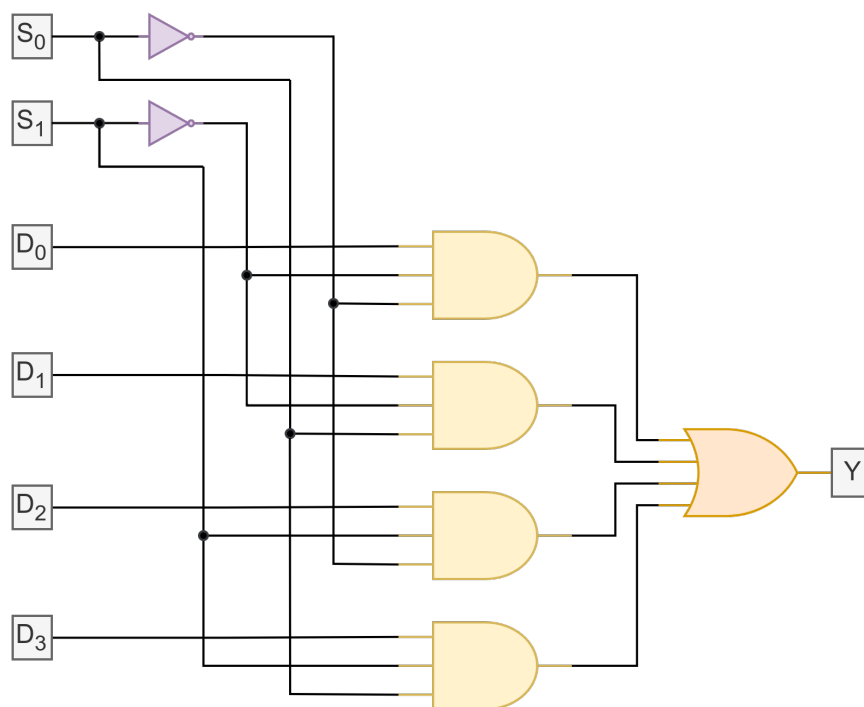
Zie de opgestelde waarheidstabel van de 4-bit multiplexer (figuur 5):

S_1	S_0	Input Selected
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

6.2 Herleid vanuit de opgestelde waarheidstabel de bijbehorende logische uitdrukking

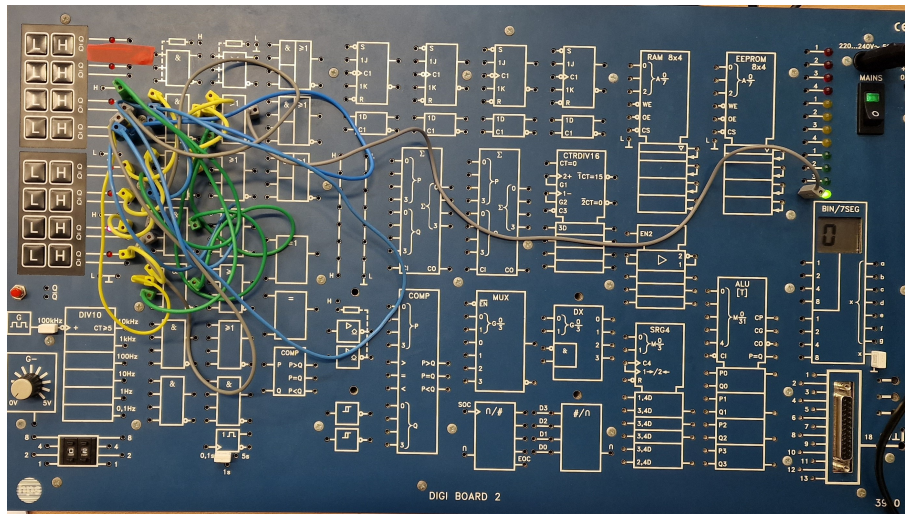
De bijbehorende logische uitdrukking: $D_0 \cdot \overline{S_1} \cdot \overline{S_0} + D_1 \cdot \overline{S_1} \cdot S_0 + D_2 \cdot S_1 \cdot \overline{S_0} + D_3 \cdot S_1 \cdot S_0$.

6.3 Teken de logische schakeling die deze uitdrukking representeert



Figuur 6: Ontwerp van de logische schakeling van de 4-bit multiplexer. Deze MUX heeft twee select pennen en vier data pennen.

6.4 Bouw de schakeling op het digiboard



Figuur 7: Foto van de ontworpen logische schakeling van de multiplexer dat gebouwd is op het digiboard.

6.5 Controleer de werking van de logische schakeling

Tijdens het practicum is het ontworpen logische schakeling gebouwd op het digiboard en werkte de multiplexer succesvol. Zie de resultaten hieronder:

- De data-uitgang y is alleen gelijk aan D_0 wanneer $S_1 = 0$ en $S_0 = 0$;
- De data-uitgang y is alleen gelijk aan D_1 wanneer $S_1 = 0$ en $S_0 = 1$;
- De data-uitgang y is alleen gelijk aan D_2 wanneer $S_1 = 1$ en $S_0 = 0$;
- De data-uitgang y is alleen gelijk aan D_3 wanneer $S_1 = 1$ en $S_0 = 1$.

Dit komt overeen met de eerder opgestelde waarheidstabel en de logische uitdrukking: $D_0 \cdot \overline{S_1} \cdot \overline{S_0} + D_1 \cdot \overline{S_1} \cdot S_0 + D_2 \cdot S_1 \cdot \overline{S_0} + D_3 \cdot S_1 \cdot S_0$.

7 Combinatorische logische functie: BCD-encoder

In de tweede opdracht zijn de volgende deelvragen onderzocht en beantwoord over de BCD-encoder:

1. Stel een BCD-tabel op voor decimale getallen 0 t/m 9;
2. Herleid vanuit deze opgestelde tabel de bijbehorende logische uitdrukking;
3. Teken de logische schakeling die deze uitdrukking representeert;
4. Bouw de schakeling op het digiboard;
5. Controleer de werking van de logische schakeling door de BCD-uitgangen aan te sluiten op de BIN/7SEG decoder.

7.1 Stel een BCD-tabel op voor decimale getallen 0 t/m 9

<i>Decimaal</i>	<i>Binary-Coded-Decimal</i>
	$\overline{A B C D}$
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

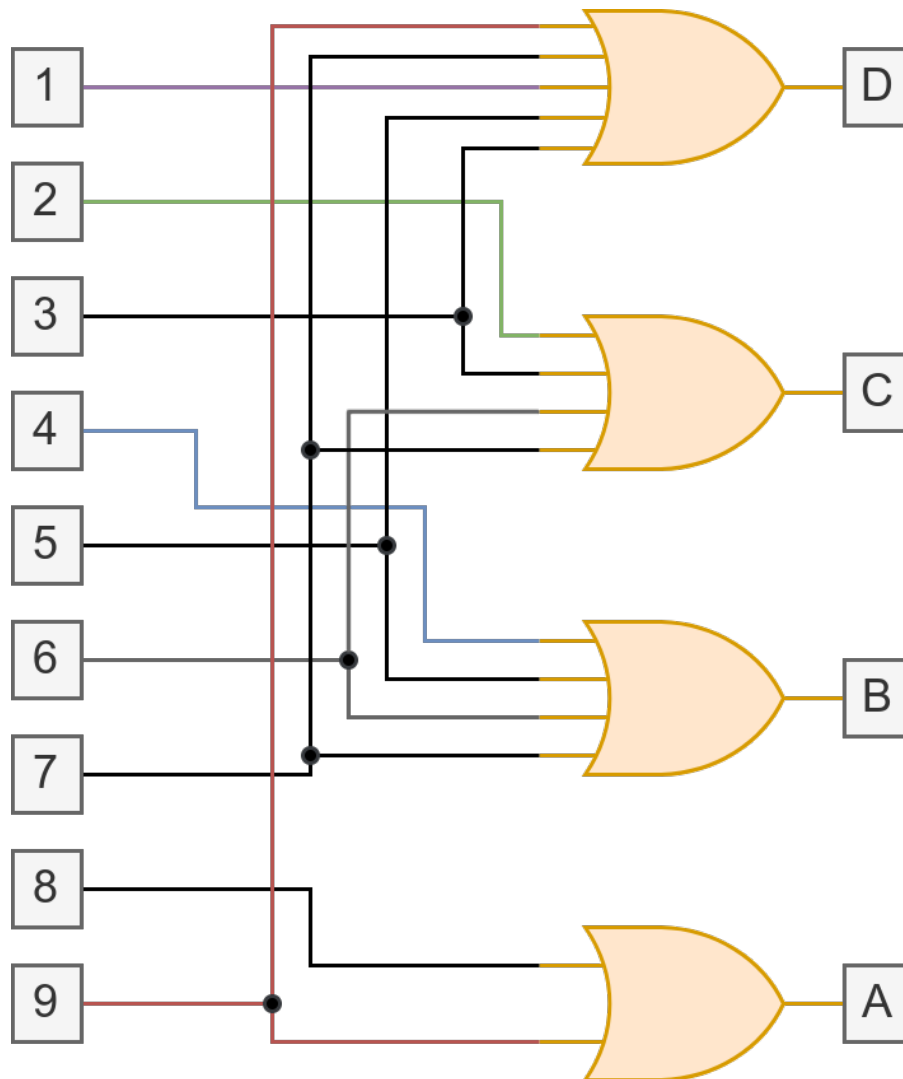
7.2 Herleid vanuit deze opgestelde tabel de bijbehorende logische uitdrukking

De bijbehorende logische uitdrukkingen voor de BCD-encoder:

- $A (X_3) = A \overline{B} \overline{C} \overline{D} + A \overline{B} \overline{C} D;$
- $B (X_2) = \overline{A} B \overline{C} \overline{D} + \overline{A} B \overline{C} D + \overline{A} B C \overline{D} + \overline{A} B C D;$
- $C (X_1) = \overline{A} \overline{B} C \overline{D} + \overline{A} \overline{B} C D + \overline{A} B C \overline{D} + \overline{A} B C D;$
- $D (X_0) = \overline{A} \overline{B} \overline{C} D + \overline{A} \overline{B} C D + \overline{A} B \overline{C} D + \overline{A} B C D + A \overline{B} \overline{C} D.$

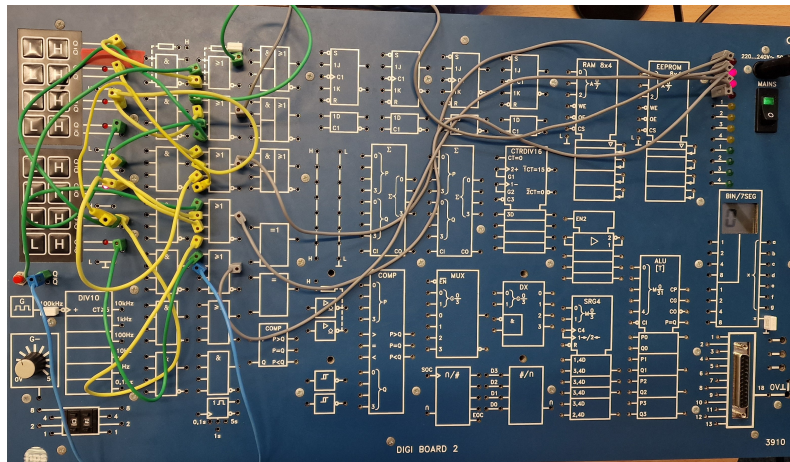
7.3 Teken de logische schakeling die deze uitdrukking representeert

Zie de ontworpen logische schakeling hieronder:



Figuur 8: Logische schakeling van de BCD-encoder. Het getal '0' is hierbij niet meegenomen, omdat bij dat getal er geen LED aangaat (binair: 0000).

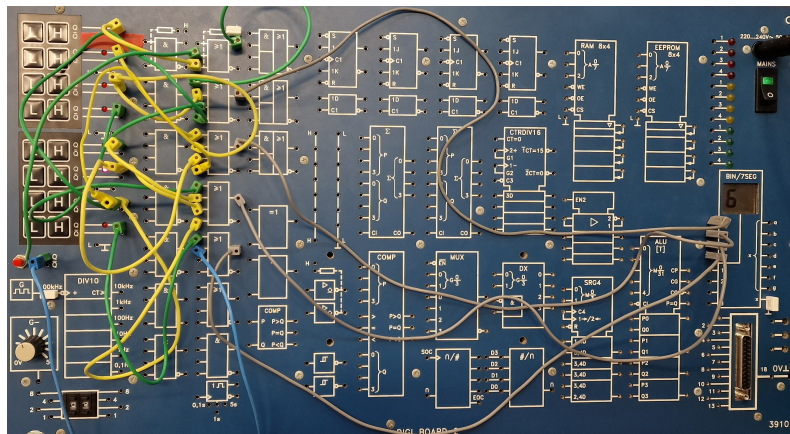
7.4 Bouw de schakeling op het digiboard



Figuur 9: Foto van de gebouwde logische schakeling van de BCD-encoder.

7.5 Controleer de werking van de logische schakeling door de BCD-uitgangen aan te sluiten op de BIN/7SEG decoder

Tijdens het practicum is de output aangesloten aan de BIN/7SEG decoder en werkt het allemaal zoals het zou moeten, zie de afbeelding hieronder:



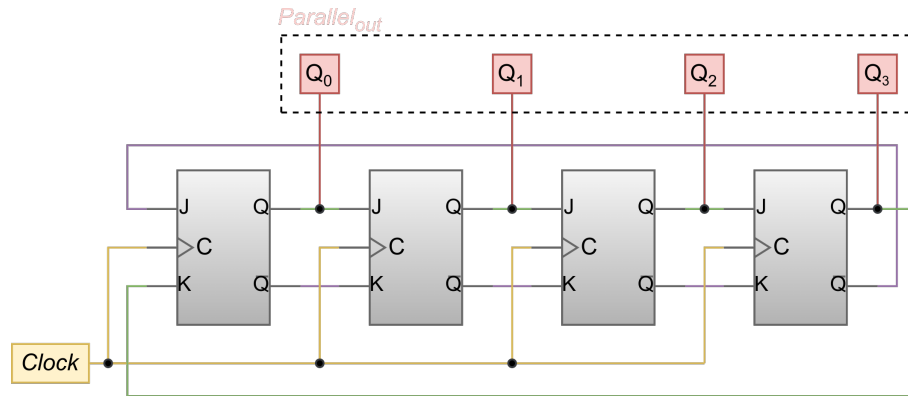
Figuur 10: Foto van de gebouwde logische schakeling van de BCD-encoder waarbij de output is aangesloten aan een BIN/7SEG decoder.

8 Teller ($2N$ -toestanden)

In de derde opdracht zijn de volgende deelvragen onderzocht en beantwoord over de teller met $2N$ -toestanden:

1. Ontwerp met behulp van 4 J-K-flip-flops een teller met $2N$ toestanden;
2. Stel aan de hand van het ontwerp, de bijbehorende toestandstabel op;
3. Hoe wordt deze teller ook wel genoemd;
4. Bouw deze schakeling op het digiboard;
5. Sluit de uitgang van elke flip-flop aan op een LED;
6. Sluit op de klokingang van de J-K-flip-flops een blokgolf met een frequentie van 1 Hz aan (gebruik hiervoor de DIV 10 generator die zich op het digiboard bevindt);
7. Controleer of de schakeling alle toestanden doorloopt;
8. Eigenschappen en tijdvolgordediagram.

8.1 Ontwerp met behulp van 4 J-K-flip-flops een teller met $2N$ toestanden



Figuur 11: Logische schakeling van de $2N$ Teller.

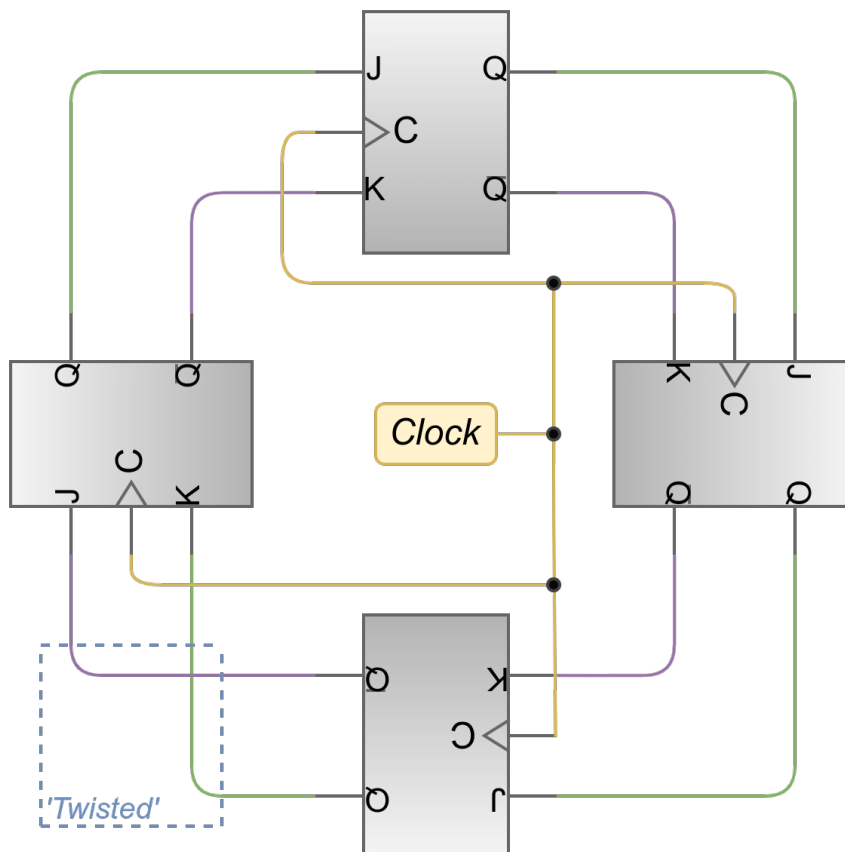
8.2 Stel aan de hand van het ontwerp, de bijbehorende toestandstabel op

De Johnsonsteller heeft $2N$ toestanden, dat betekent dat er maximaal acht mogelijke toestanden zijn (0 tot en met 7) [9]. Zie hieronder de bijbehorende toestandstabel van de twisted ring counter (Johnsonsteller):

Twisted ring counter				
Toestand	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
0	0	0	0	0

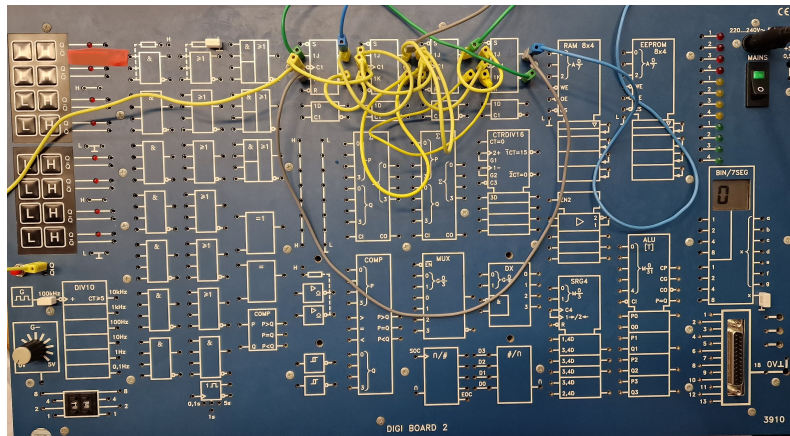
8.3 Hoe wordt deze teller ook wel genoemd

Deze teller met $2N$ toestanden wordt ook wel een “Johnson counter” of “Twisted ring counter” genoemd. De ‘Q’ output van de laatste J-K-flip-flop is verbonden aan K van de eerste J-K-flip-flop en hetzelfde principe geldt ook voor \overline{Q} naar J. Dit is de reden waarom het een “twisted” ring counter wordt genoemd, maar ook omdat het een ring structuur heeft. Zie de afbeelding hieronder van de twisted ring counter:



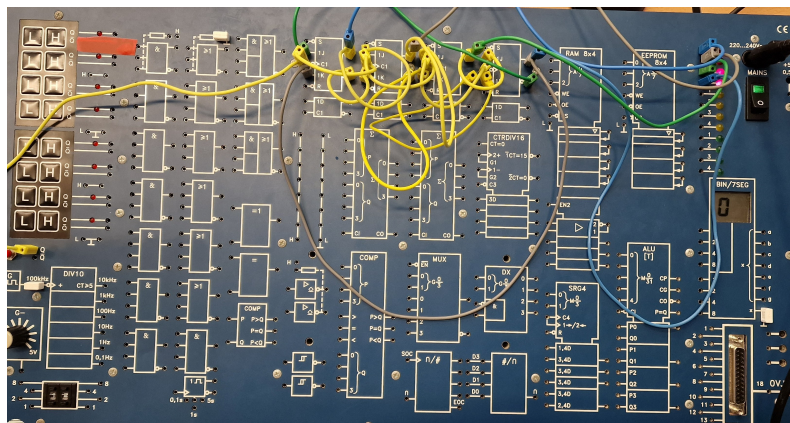
Figuur 12: Logische schakeling van twisted ring counter.

8.4 Bouw deze schakeling op het digiboard



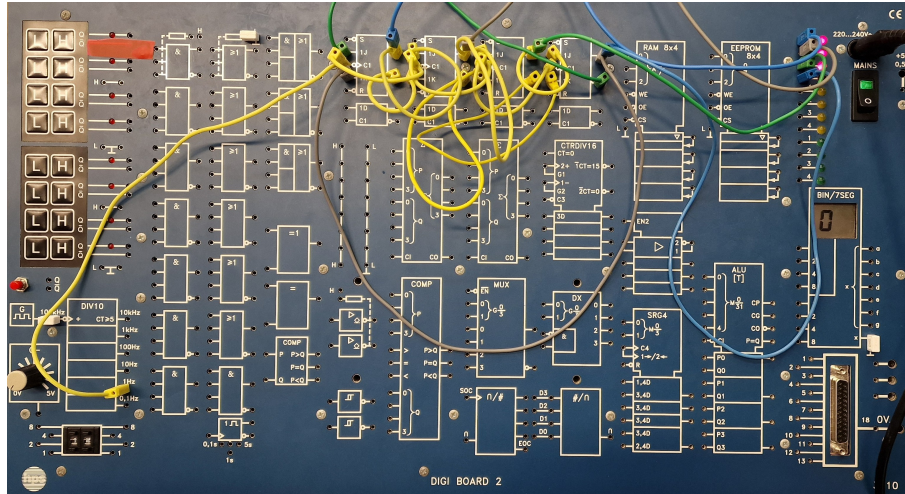
Figuur 13: Gebouwde logische schakeling van twisted ring counter.

8.5 Sluit de uitgang van elke flip-flop aan op een LED



Figuur 14: Gebouwde logische schakeling van twisted ring counter. De output snoeren zijn aangesloten aan LED's.

8.6 Sluit op de klokingang van de J-K-flip-flops een blokgolf met een frequentie van 1 Hz aan



Figuur 15: Gebouwde logische schakeling van twisted ring counter.

De output snoeren zijn aangesloten aan LED's en de klokingang is aangesloten aan de DIV10 blokgolf-generator.

8.7 Controleer of de schakeling alle toestanden doorloopt

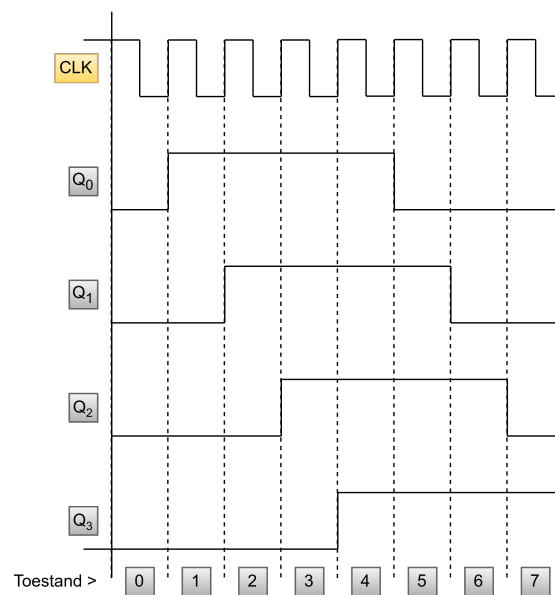
Alle toestand zijn gecontroleerd tijdens het practicum en kwamen overeen met de toestanden die zijn aangegeven in H8.2.

8.8 Eigenschappen en tijdvolgordediagram

De Johnson teller wordt in veel digitale systemen toegepast vanwege de volgende unieke eigenschappen:

- De Johnson teller verbruikt minimale energie, omdat er bij elke klokpuls maar een J-K-flip-flop veranderd van toestand;
- De Johnson teller heeft $2N$ toestanden. Dit betekent dat er acht verschillende combinaties mogelijk zijn [10].

Zie hieronder de tijdvolgordediagram van de Johnson teller:



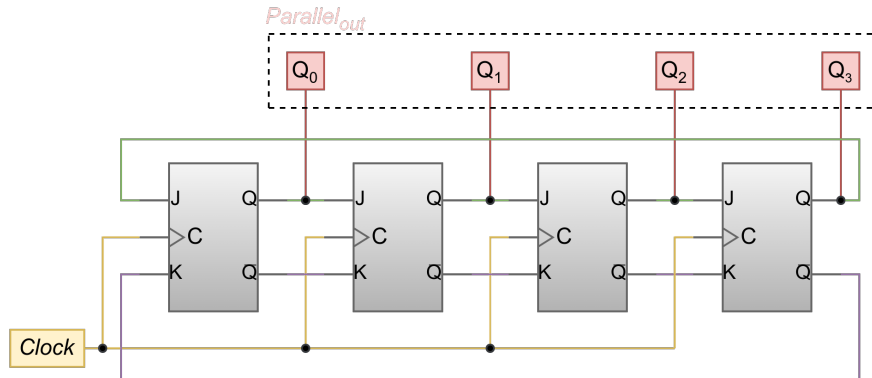
Figuur 16: Tijdvolgordediagram van de Johnson teller.

9 Teller (N-toestanden)

In de vierde opdracht zijn de volgende deelvragen onderzocht en beantwoord over de teller met N-toestanden:

1. Ontwerp met behulp van 4 J-K-flip-flops een teller met N-toestanden;
2. Stel aan de hand van het ontwerp, de bijbehorende toestandstabel op;
3. Hoe wordt deze teller ook wel genoemd;
4. Bouw deze schakeling op het digiboard;
5. Sluit de uitgang van elke flip-flop aan op een LED;
6. Sluit op de klokingang van de J-K-flip-flops een blokgolf met een frequentie van 1 Hz aan (gebruik hiervoor de DIV 10 generator die zich op het digiboard bevindt);
7. Wat gebeurt er;
8. Hoe zorgen we ervoor dat de teller start;
9. Controleer of de schakeling alle toestanden doorloopt;
10. Eigenschappen en tijdvolgordediagram.

9.1 Ontwerp met behulp van 4 J-K-flip-flops een teller met N-toestanden



Figuur 17: Logische schakeling van de N Teller.

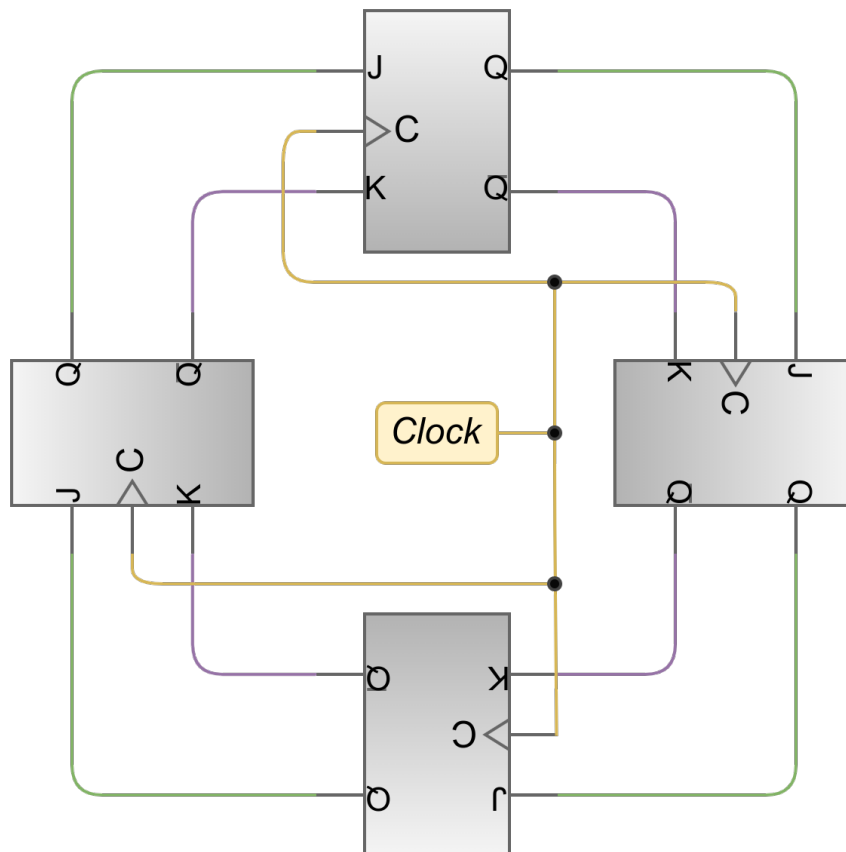
9.2 Stel aan de hand van het ontwerp, de bijbehorende toestandstabel op

De ringteller heeft N toestanden, dat betekent dat er maximaal vier mogelijke toestanden zijn (0 tot en met 3) [9]. Zie hieronder de bijbehorende toestandstabel van de (straight) ringteller:

Straight ring counter				
Toestand	Q_0	Q_1	Q_2	Q_3
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
0	1	0	0	0

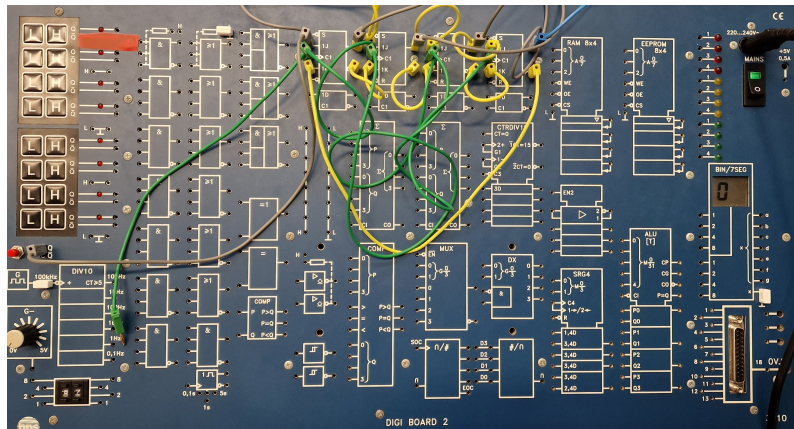
9.3 Hoe wordt deze teller ook wel genoemd

Deze teller wordt ook wel de “straight ring counter” genoemd. Deze naam is gegeven omdat deze ring teller geen “twist” heeft, maar omdat alle flip-flops op dezelfde wijze met input en output aangesloten zijn.



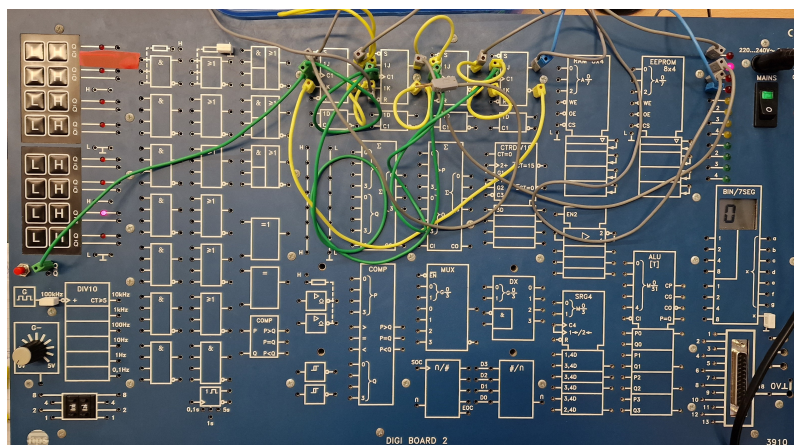
Figuur 18: Logische schakeling van straight ring counter.

9.4 Bouw deze schakeling op het digiboard



Figuur 19: Gebouwde logische schakeling van straight ring counter.

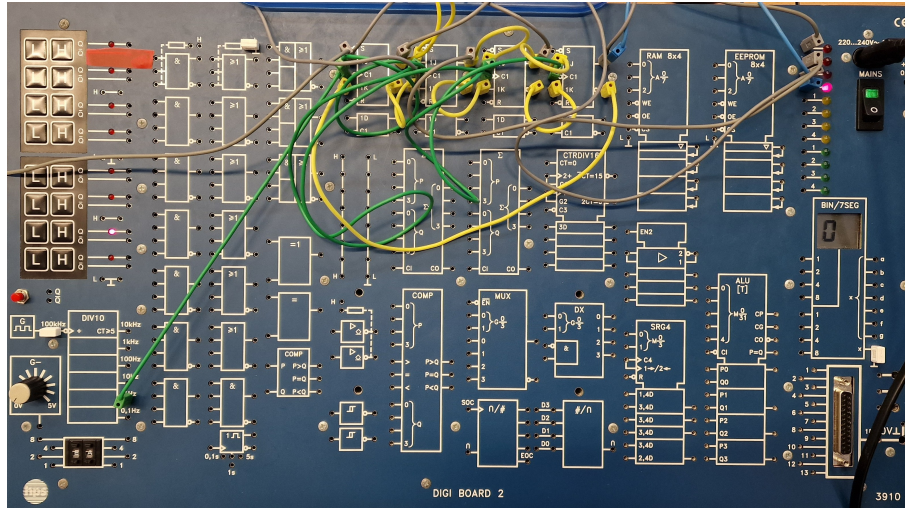
9.5 Sluit de uitgang van elke flip-flop aan op een LED



Figuur 20: Gebouwde logische schakeling van straight ring counter.

Hierbij zijn de output snoeren aangesloten aan LED's.

9.6 Sluit op de klokingang van de J-K-flip-flops een blokgolf met een frequentie van 1 Hz aan



Figuur 21: Gebouwde logische schakeling van straight ring counter.

Hierbij zijn de output snoeren aangesloten aan LED's. Ook is de klokingang aangesloten aan de DIV10 blokgolf-generator met een frequentie van 1 Hz.

9.7 Wat gebeurt er

De straight ring counter gaat met een frequentie van 1 Hz door alle toestanden heen.

9.8 Hoe zorgen we ervoor dat de teller start

Eerst moet de eerste J-K-flip-flop geladen worden met een korte actief-hoge puls, daarna kan de klok geactiveerd worden. Nu zal de ring teller ervoor zorgen dat die actief-hoge puls doorgevoerd zal worden naar de volgende toestand, zoals beschreven in de toestandstabel.

9.9 Controleer of de schakeling alle toestanden doorloopt

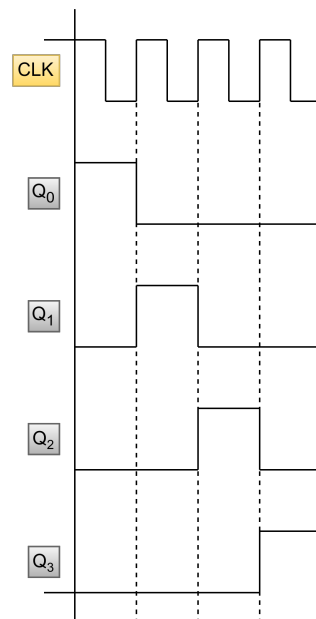
De toestanden waren tijdens het practicum gecontroleerd en werkte de teller succesvol. Alle toestanden kwamen precies overeen zoals gerepresenteerd in H9.2.

9.10 Eigenschappen en tijdvolgordediagram

De ringteller heeft de volgende eigenschappen:

- De ringteller kan niet zelf starten. Dit houdt in dat er in eerste instantie een actief-hoge puls geladen moet worden door een van de J-K-flip-flop's;
- De ringteller heeft N toestanden. Dit betekent dat er vier verschillende combinaties mogelijk zijn [10].

Zie hieronder de tijdvolgordediagram van de ringteller:



Figuur 22: Tijdvolgordediagram van de ringteller.

10 Conclusie

In dit verslag is er onderzoek gedaan naar de werking van de multiplexer, de BCD-encoder, straight ring counter en de twisted ring counter (Johnson counter). Daarbij was ook een probleemstelling geïntroduceerd in H2 en de bijbehorende doelstellingen, zie hieronder:

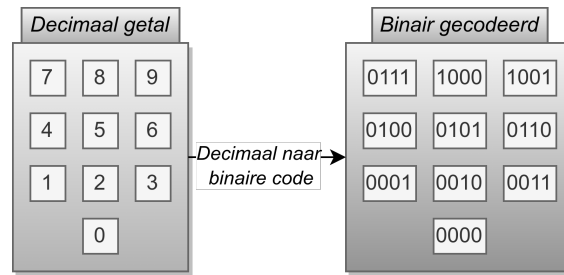
1. Inzicht te verkrijgen in logische functies die ten grondslag liggen aan de digitale techniek;
2. Waarnemingen te verrichten aan de bijbehorende eenvoudige logische schakelingen;
3. De bijbehorende meetgegevens en waarheidstabellen op een juiste wijze te verkrijgen en te interpreteren;
4. De vergaarde gegevens te verwerken tot een verslag.

De probleemstelling was:

- *“Inzicht krijgen in de werking van een aantal complexe logische schakelingen en daarna deze op een efficiënte wijze te ontwerpen en te bouwen?”*

Bij de eerste practicumopdracht *“Combinatorische logische functie: MUX”* werd geëxperimenteerd met een 4-bit multiplexer. Hierbij was de opdracht om een logische schakeling te ontwerpen en te bouwen dat ervoor zorgt dat er 4-bits aan data-lijnen gekozen kunnen worden door middel van 2-bits data-select-lijnen. Om dat te kunnen doen is er gebruik gemaakt van een waarheidstabel en een SOP-expressie. De bijbehorende logische schakelingen zijn allemaal gebouwd, getest en gecontroleerd op de werking ervan.

Vervolgens is er in de tweede practicumopdracht *“Combinatorische logische functie: BCD-encoder”* geëxperimenteerd met een Binary-Coded-Decimal encoder. Deze encoder heeft als functie dat de decimale input met getallen van 0 tot en met 9 wordt omgezet in binaire code, dus daarom is het daarnaar vernoemd. Tijdens het testen van de logische schakeling kwamen we erachter dat het getal ‘0’ niet een input knop heeft, omdat het getal ‘0’ in binair “0000” is. De output was aangesloten aan de BCD-decoder van het digiboard en die vertaalde dat gelukkig automatisch naar ‘0’. Voor de rest werkte de logische schakelingen zoals vanuit de theorie verwacht was, en waren er geen andere problemen. De werking van de encoder is hieronder nog gerepresenteerd:



Figuur 23: Abstracte blokschema dat de werking van de BCD-encoder representeert.

In de derde practicumopdracht “*Teller (2N-toestanden)*” is er geëxperimenteerd met een twisted ringteller ook wel bekend als de Johnson counter. Deze schakeling maakt gebruik van vier J-K-flip-flop’s: de klokingangen zijn allemaal aangesloten aan dezelfde klokgenerator, de ‘Q’ output van de laatste J-K-flip-flop is verbonden aan K van de eerste J-K-flip-flop en hetzelfde principe geldt ook voor \overline{Q} naar input J. De Johnson teller wordt in veel digitale systemen toegepast vanwege de volgende unieke eigenschappen:

- De Johnson teller verbruikt minimale energie, omdat er bij elke klokpuls maar een J-K-flip-flop veranderd van toestand;
- De Johnson teller heeft $2N$ toestanden. Dit betekent dat er acht verschillende combinaties mogelijk zijn.

De Johnson teller schakelt dus snel, omdat de teller werkt in een éénwisselcode. De alternatieve naam “twisted ring counter” komt van het feit dat de logische schakelingen een “ring” als structuur heeft. Daarnaast bevat de schakeling een “twist”, dus vandaar “twisted ring counter”.

Bij de laatste opdracht “*Teller (N-toestanden)*” is er geëxperimenteerd met een straight ring counter ook wel bekend als de ringteller. Deze schakeling maakt gebruik van vier J-K-flip-flop’s: de klokingangen zijn allemaal aangesloten aan dezelfde klokgenerator, de J ingangen en uitgangen zijn allemaal doorgelust aan elkaar en de K ingangen en uitgangen zijn ook allemaal doorgelust aan elkaar. De ringteller heeft de volgende eigenschappen:

- De ringteller kan niet zelf starten. Dit houdt in dat er in eerste instantie een actief hoge puls geladen moet worden door een van de J-K-flip-flop’s;
- De ringteller heeft N toestanden. Dit betekent dat er vier verschillende combinaties mogelijk zijn.

Deze ringteller wordt voornamelijk toegepast in sequentiële systemen, omdat deze tellers tijdbepalende pulsen kunnen genereren.

De bovenstaande probleemstelling is dan ook op een succesvolle wijze opgelost door:

1. Eerst een waarheidstabel te maken van de toegepaste logische schakeling. Hiermee worden alle Booleaanse mogelijkheden op een overzichtelijke manier gepresenteerd;
2. Als tweede stap is op basis van de waarheidstabel een eenvoudige Booleaanse expressie opgesteld;
3. Als derde stap is een Karnaugh diagram gemaakt. Daarmee konden er allerlei groepjes gevormd worden van binaire getallen om de minimalisatie mogelijk te maken;
4. Tenslotte werd een logische schakeling ontworpen en is deze gebouwd op het digiboard. Op deze gestructureerde manier is er op een effectieve manier een toegepaste logische schakeling ontworpen en gebouwd!

Dus er kan geconcludeerd worden dat de vier doelstellingen van dit practicumverslag met succes gerealiseerd zijn!

11 Bronvermelding

Referenties

- [1] HPS SystemTechnik GmbH, “3910 DIGI BOARD 2 Datasheet,” <https://hps-systemtechnik.com/wp-content/uploads/content/english/3910.pdf>, 2014, accessed on [29-4-2023].
- [2] “Logische basispoorten,” Website, accessed: [30-4-2023]. [Online]. Available: <https://electricalengineering123.com/logic-gates-xor-nand-nor-and-gate-truth-table/>
- [3] E. L. Post, “Introduction to a general theory of elementary propositions,” *American Journal of Mathematics*, vol. 43, no. 3, p. 163, Jul. 1921. [Online]. Available: <https://doi.org/10.2307/2370324>
- [4] iLogic, “3.2.5 truth tables for arguments,” bit.ly/3Vr0aSf, 2008, accessed: [2-5-2023].
- [5] “Booleaanse wetten,” Website, accessed: [7-5-2023]. [Online]. Available: <https://www.mi.mun.ca/users/cchaulk/misc/boolean.htm>
- [6] A. devices, “Quick-start: Driving 7-segment displays with the max6954,” <https://www.analog.com/en/design-notes/quickstart-driving-7segment-displays-with-the-max6954.html>, accessed: [30-5-2023].
- [7] T. H. Flowers, “The design of colossus (foreword by howard campagne),” *IEEE Annals of the History of Computing*, vol. 5, no. 3, pp. 239–252, Jul. 1983. [Online]. Available: <https://doi.org/10.1109/mahc.1983.10079>
- [8] “Hps systemtechnik,” Website, accessed: [29-4-2023]. [Online]. Available: <https://hps-systemtechnik.com>
- [9] B. Holdsworth and C. Woods, *Digital Logic Design*. Elsevier Science, 2002. [Online]. Available: <https://books.google.nl/books?id=o7enSwSVvgYC>
- [10] docplayer en Gerda van der Horst, “Voorstelling van de werking van een schuifregister,” <https://docplayer.nl/8228989-Fig-6-1-voorstelling-van-de-werking-van-een-schuifregister.html>, accessed: [3-6-2023].
- [11] HPS SystemTechnik GmbH, “DIGI BOARD 2 (Type 3530) Datasheet,” <https://hps-systemtechnik.com/wp-content/uploads/content/english/3530.pdf>, 2014, accessed on [29-4-2023].
- [12] I. M. Copi, C. Cohen, and K. McMahon, *Introduction to Logic*. Routledge, Sep. 2016. [Online]. Available: <https://doi.org/10.4324/9781315510897>

- [13] M. Karnaugh, “The map method for synthesis of combinational logic circuits,” *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, vol. 72, no. 5, pp. 593–599, 1953. [Online]. Available: <https://doi.org/10.1109/tce.1953.6371932>