

Digitale Techniek - Practicum 2: Functies van combinatorische logica

Semih Can Karakoç (695258)

8 mei 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	Het digiboard en bijbehorende IC-board	8
5	Methode en aanpak	11
6	HALF-ADDER	13
6.1	Ontwerp een HALF-ADDER gebouwd met alleen basispoorten	14
6.2	Bouw de logische schakeling op het digiboard	14
6.3	Ga de werking na en stel de bijbehorende waarheidstabel op	15
6.4	Bouw nu een HALF-ADDER met behulp van een XOR en een AND-poort. Vergelijk de werking met de voorgaande schakeling	16
7	FULL-ADDER	17
7.1	Ontwerp een FULL-ADDER bestaande uit twee HALF-ADDER's en een OR-poort	18
7.2	Bouw de logische schakeling op het digiboard	18
7.3	Ga de werking na en stel de bijbehorende waarheidstabel op	19
7.4	Bouw de schakeling nu met behulp van 2 XOR-poorten, 2 AND-poorten en 1 OR-poort en controleer de werking.	20
7.5	Maak een waarheidstabel	20
8	6-bits opteller	21
9	Vergelijking van binaire getallen	23
9.1	Ontwerp een schakeling waarmee twee bits kunnen worden getest op ongelijkheid. Deze schakeling mag alleen maar bestaan uit NAND's	24
9.2	Bouw de logische schakeling op het digiboard	24
9.3	Ga de werking na en stel de bijbehorende waarheidstabel op	25
10	Combinatorische logische schakeling	26
10.1	Bepaal de bij deze schakeling behorende SOP-uitdrukking	27

10.2 Ontwerp de bijbehorende logische schakeling	27
10.3 Bouw de logische schakeling op het digiboard	28
10.4 Controleer de werking van deze schakeling. Maak voor de niveausen- sors gebruik van de schakelaars op het digiboard	28
11 Conclusie	29
12 Bronvermelding	31

1 Inleiding

Tijdens dit practicum is er geëxperimenteerd en onderzocht naar de werking van de HALF-ADDER, FULL-ADDER, 6-BIT opteller, 1-BIT vergelijker en een combinatorische logische schakeling. 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 vijf experimenten uitgevoerd. Daarbij zijn de bijbehorende deelopdrachten uitgewerkt in dit verslag.

Dit verslag zal zich richten op de volgende onderwerpen:

1. Logische schakelingen lezen, bouwen en ontwerpen;
2. HALF-ADDER en FULL-ADDER;
3. 1-BIT COMPARATOR;
4. Waarheidstabellen;
5. Gedrag van logische poorten;
6. Booleaanse algebra.

Het verslag zal de lezer eerst de theorie laten behandelen en daarna laten ingaan op de uitgevoerde experimenten, de bijbehorende deelopdrachten, de probleemstelling en tenslotte eindigen met een concrete eindconclusie.

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

2 Probleem- en doelstelling

Dit verslag bevat een algemene probleemstelling en dat is:

- *“Hoe kunnen we op een effectieve manier diverse logische schakelingen ontwerpen en bouwen door het gebruik van zowel eenvoudige als geavanceerde logische poorten?”*

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 11: *“Conclusie”* zal de gestelde algemene probleemstelling beantwoord worden.

3 Practicumbenodigdheden

Hieronder zijn de benodigdheden benoemd die essentieel zijn voor tijdens het practicum:

1. Digiboard: het 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. IC-board: het IC bord is de uitbreiding van de digiboard. Op het IC bord kun je allerlei IC's naar keuze erin klemmen en gemakkelijk alles op een juiste manier aansluiten [2];
3. Verbindingsnoeren: deze snoeren zijn nodig om verbinding te maken tussen allerlei elektronische componenten;
4. 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 \bar{A} (A niet) [3].

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 [4]. 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 [5].

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 $\overline{A + B} = \bar{A} \cdot \bar{B}$;
- Dubbele negatie: $\overline{\bar{A}} = A$ [6].

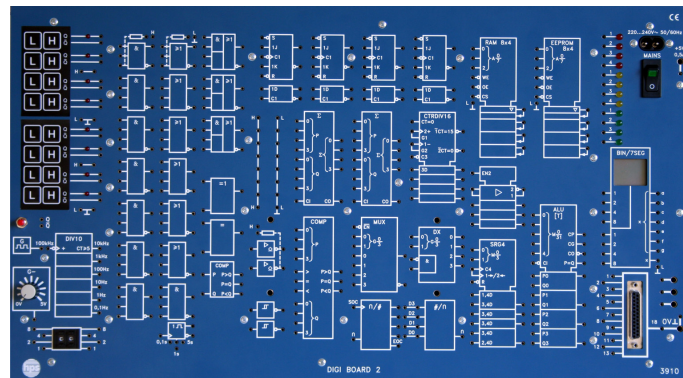
4.4 Het digiboard en bijbehorende IC-board

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 [7]. 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;

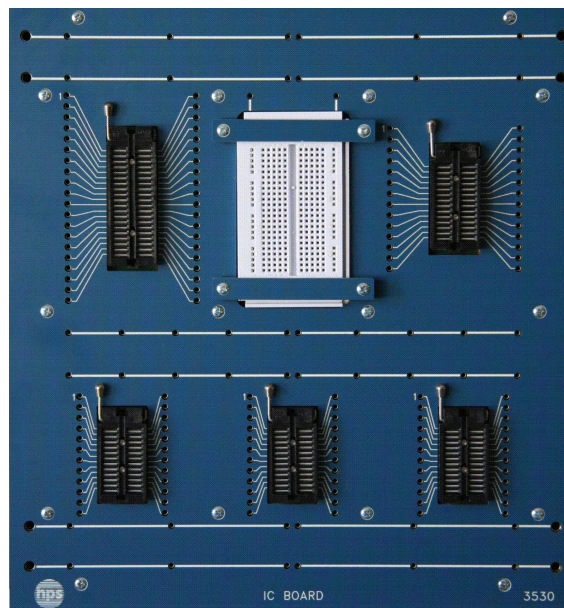
12. 1-bit comparator;
13. 4-bit comparator;
14. 4 JK-flipflops, can also be used as RS flipflops;
15. 4 D-flipflops;
16. 2 adders (4-bit), with input and output carry;
17. Monoflop, settable times: 0.1s; 1s; 5s;
18. Multiplexer, 4 channels;
19. Demultiplexer, 4 channels;
20. Shift register (4-bit), parallel 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 de digiboard:



Figuur 1: Foto van de “Type 3910 DIGI BOARD 2”.

Daarnaast heeft *hps SystemTechnik* meerdere extensie borden gefabriceerd voor de digiboard. Een van deze borden is de “Type 3530 IC-BOARD”. In het practicum is er gewerkt met deze IC-board, omdat het bord het heel makkelijk maakt om logische schakeling te bouwen met IC's. Zie de afbeelding hieronder voor een representatie van de “Type 3530 IC BOARD” [2]:



Figuur 2: Foto van de “Type 3530 IC BOARD”.

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 de 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;

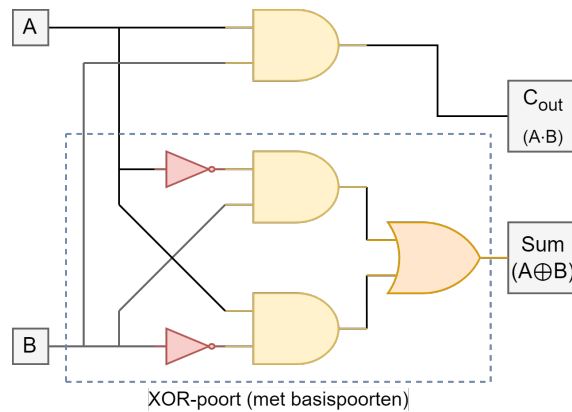
- In de methode geef je uitleg over hoe je alles hebt aangepakt en uitgewerkt hebt, zodat de lezer het practicum mogelijk zelf zou kunnen uitvoeren;
- De resultaten/antwoorden splits je op in secties net zoals de vragen zijn gesteld in het practicum bestand. Deze resultaten werk je netjes uit en zet 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 HALF-ADDER

In de eerste opdracht zijn de volgende deelvragen onderzocht en beantwoord over de HALF-ADDER:

1. Ontwerp een HALF-ADDER. Voor het ontwerp mag alleen gebruik worden gemaakt van basispoorten;
2. Bouw de logische schakeling op het digiboard;
3. Ga de werking na en stel de bijbehorende waarheidstabel op;
4. Bouw nu een HALF-ADDER met behulp van een XOR en een AND-poort. Vergelijk de werking met de voorgaande schakeling.

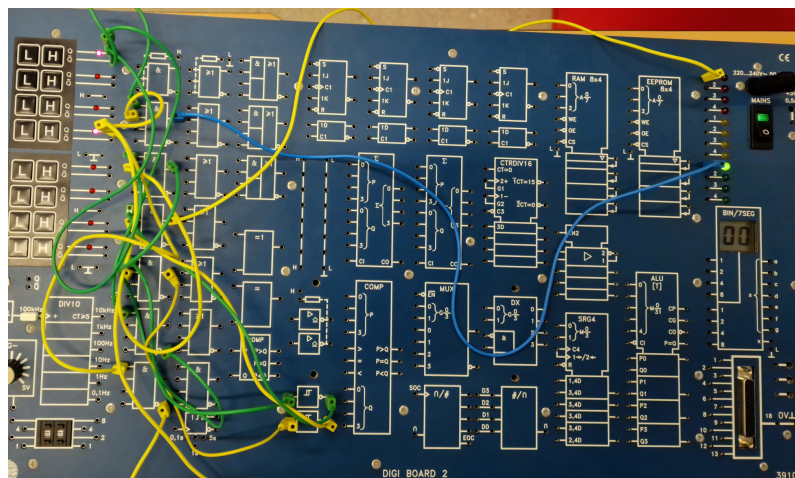
6.1 Ontwerp een HALF-ADDER gebouwd met alleen basispoorten



Figuur 3: Ontwerp van een HALF-ADDER dat bestaat uit alleen de basispoorten (= AND, OR en NOT-poorten).

6.2 Bouw de logische schakeling op het digiboard

Zie de foto hieronder:



Figuur 4: Foto van een HALF-ADDER dat bestaat uit alleen de basispoorten (= AND, OR en NOT-poorten).

6.3 Ga de werking na en stel de bijbehorende waarheidstabel op

Een HALF-ADDER is een logische schakeling die zowel twee inputs als outputs heeft. De HALF-ADDER bestaat uit een AND-poort en een XOR-poort. De AND-poort wordt in deze logische schakeling ook wel de “Carry Out” (C_{out}) genoemd. De XOR-poort wordt in deze logische schakeling ook wel de “Som” genoemd. Bovendien kun je met twee HALF-ADDER’s kun je een FULL-ADDER maken, daar gaan we later in het verslag op in.

De werking van de HALF-ADDER is als volgt:

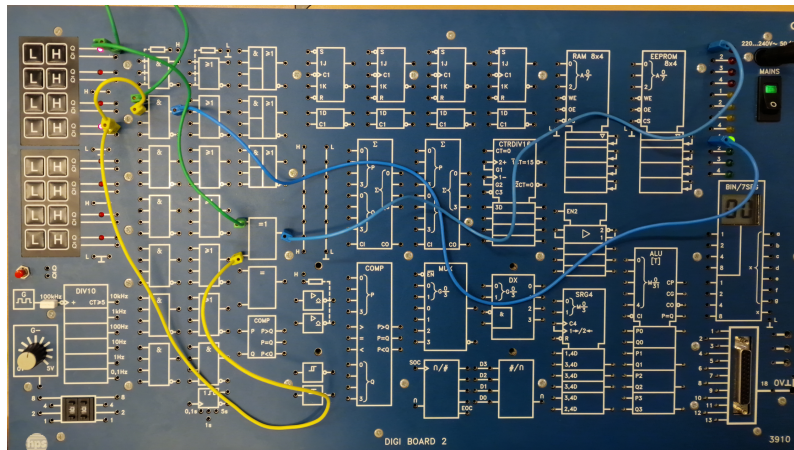
- Als beide inputs ‘0’ zijn, dan zijn de som (Σ) en C_{out} beide ‘0’;
- Als één input ‘1’ is, en de andere input ‘0’ is, dan is de som (Σ) = ‘1’ en de C_{out} ‘0’;
- Als beide inputs ‘1’ zijn, dan zijn de som (Σ) en C_{out} beide ‘1’.

Zie hieronder de bijbehorende waarheidstabel, hierbij is $A \cdot B$ de Carry Out, en is $\overline{A} \cdot B + A \cdot \overline{B}$ de som:

A	B	\overline{A}	\overline{B}	$A \cdot B$	$\overline{A} \cdot B$	$A \cdot \overline{B}$	$\overline{A} \cdot B + A \cdot \overline{B}$
1	1	0	0	1	0	0	0
1	0	0	1	0	0	1	1
0	1	1	0	0	1	0	1
0	0	1	1	0	0	0	0

6.4 Bouw nu een HALF-ADDER met behulp van een XOR en een AND-poort. Vergelijk de werking met de voorgaande schakeling

Het verschil tussen de vorige en deze logische schakeling uit zich in het gebruik van hoeveelheid snoeren en logische poorten. Door de logische XOR-poort op het digiboard te gebruiken, kost het veel minder snoeren en is het veel overzichtelijker ten opzichte van de vorige HALF-ADDER opzet. Zie de foto hieronder:



Figuur 5: Foto van een HALF-ADDER dat bestaat uit een XOR-poort en een AND-poort.

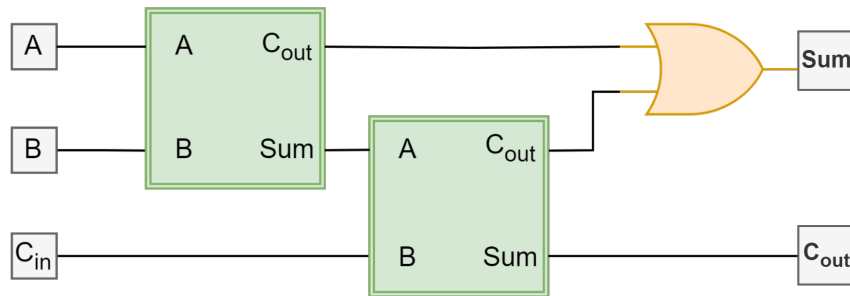
7 FULL-ADDER

In de tweede opdracht zijn de volgende deelvragen onderzocht en beantwoord over de FULL-ADDER:

1. Ontwerp een FULL-ADDER bestaande uit twee HALF-ADDER's en een OR-poort;
2. Bouw de logische schakeling op het digiboard;
3. Ga de werking na en stel de bijbehorende waarheidstabel op;
4. Bouw de schakeling nu met behulp van 2 XOR-poorten, 2 AND-poorten en 1 OR-poort en controleer de werking;
5. Maak een waarheidstabel.

7.1 Ontwerp een FULL-ADDER bestaande uit twee HALF-ADDER's en een OR-poort

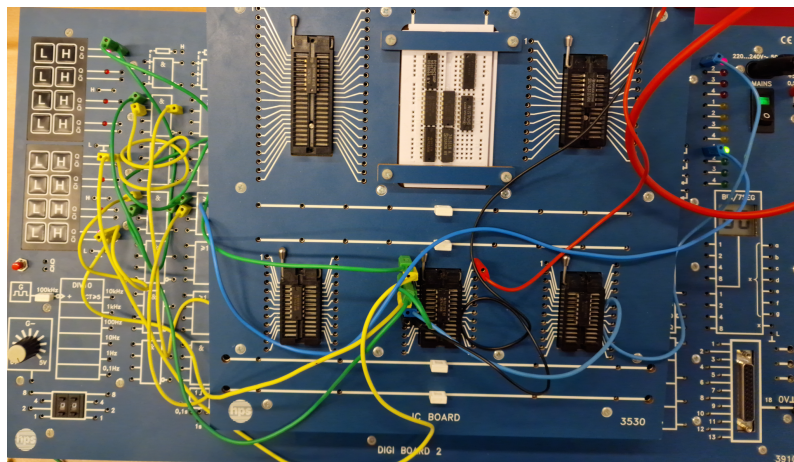
Zie de ontworpen logische schakeling hieronder:



Figuur 6: Ontwerp van een FULL-ADDER dat bestaat uit twee HALF-ADDER's en een OR-poort.

7.2 Bouw de logische schakeling op het digiboard

Zie de foto hieronder:



Figuur 7: Foto van de FULL-ADDER bestaande uit twee HALF-ADDER's en een OR-poort

7.3 Ga de werking na en stel de bijbehorende waarheidstabel op

Een FULL-ADDER is een algemene logische schakeling die wordt gebruikt om bits bij elkaar op te tellen. De “Sum” is de som van de inputs, dus als er ‘1’+‘0’ wordt uitgevoerd, wordt de sum-bit actief HOOG. De “Carry Out” wordt gebruikt wanneer er twee bits met actief HOOG signaal worden opgeteld, want dan krijg je ‘1’+‘1’ en dan is in binair ‘10’. De Carry Out draagt als het ware de ‘1’ naar de volgende bit positie.

De “Carry in” van de FULL-ADDER kan voor veel toepassingen benut worden, zoals het aftrekken van getallen.

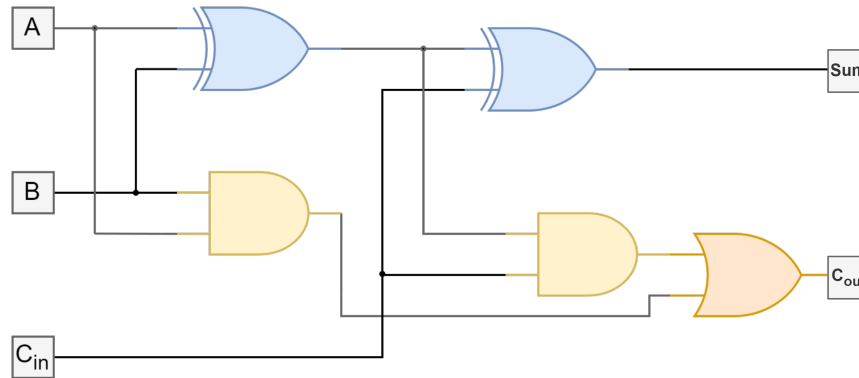
De werking van een FULL-ADDER is hieronder op een rijtje gezet:

- Als er maximaal één input gelijk is aan ‘1’, dan is de output gelijk aan de Sum;
- Als er maximaal twee inputs gelijk zijn aan ‘1’, dan is de output gelijk aan de Carry Out;
- Als alle inputs aan staan, dan staat zowel de Sum als de Carry Out aan;
- Als er geen input aan staat, dan is er ook geen output.

Zie de waarheidstabel hieronder:

A	B	C_{in}	Sum	C_{out}
1	1	1	1	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	0
0	1	1	0	1
0	1	0	1	0
0	0	1	1	0
0	0	0	0	0

7.4 Bouw de schakeling nu met behulp van 2 XOR-poorten, 2 AND-poorten en 1 OR-poort en controleer de werking.



Figuur 8: Ontwerp van een FULL-ADDER dat bestaat uit 2 XOR-poorten, 2 AND-poorten en 1 OR-poort.

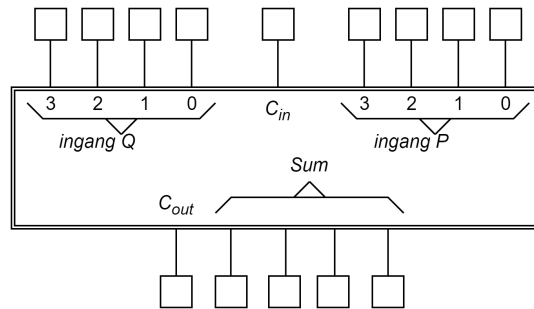
7.5 Maak een waarheidstabel

Zie de bijbehorende waarheidstabel hieronder:

A	B	C_{in}	$A \oplus B$	$A \cdot B$	$(A \oplus B) \oplus C_{in}$	$(A \oplus B) \cdot C_{in}$	$((A \oplus B) \cdot C_{in}) + (A \cdot B)$
1	1	1	0	1	1	0	1
1	1	0	0	1	0	0	1
1	0	1	1	0	0	1	1
1	0	0	1	0	1	0	0
0	1	1	1	0	0	1	1
0	1	0	1	0	1	0	0
0	0	1	0	0	1	0	0
0	0	0	0	0	0	0	0

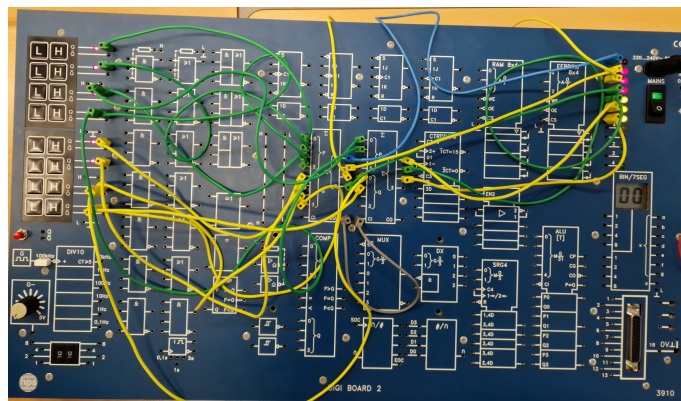
8 6-bits opteller

In deze opdracht moesten we met een “4-bit FULL-ADDER” ervoor zorgen dat we ook 6-bits op kunnen tellen, zie afbeelding 9. Op het digiboard bevinden er twee “4-bit FULL-ADDER’s”. Om ervoor te zorgen dat we 6-bits kunnen optellen, kunnen we in plaats van één “4-bit FULL-ADDER” er twee van gebruiken.

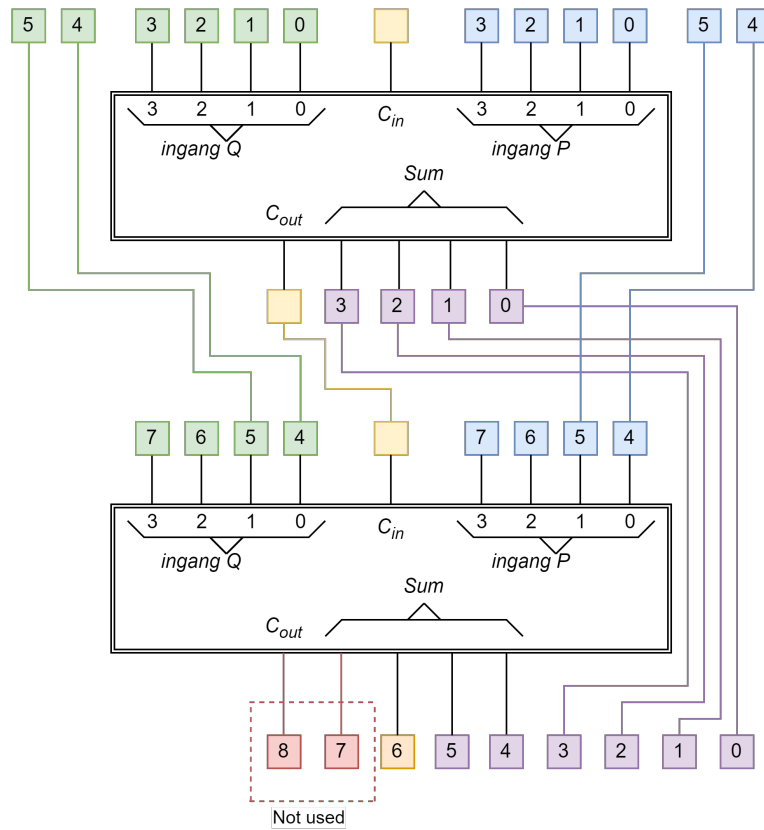


Figuur 9: Logische schakeling van de 4-bit FULL-ADDER waarvan er twee stuks op het digiboard zitten.

Om dit voor elkaar te krijgen hebben we eerst een logische schakeling ontworpen. Het idee is om de eerste 4-bit FULL-ADDER te gebruiken voor om 4-bits mee op te tellen. De tweede 4-bit FULL-ADDER gebruiken we dan om de restante 2-bits mee op te tellen en daarbij hoort ook de bijbehorende Carry Out (derde overflow-bit in de output). Ook is het belangrijk om de C_{out} van de eerste FULL-ADDER door te lussen aan de C_{in} van de tweede FULL-ADDER. De rest van de 4-bit FULL-ADDER hoeven we in principe niet te gebruiken, zie rode stuk in afbeelding 11.



Figuur 10: Ontwerp van de logische schakeling van het tank-alarm systeem.



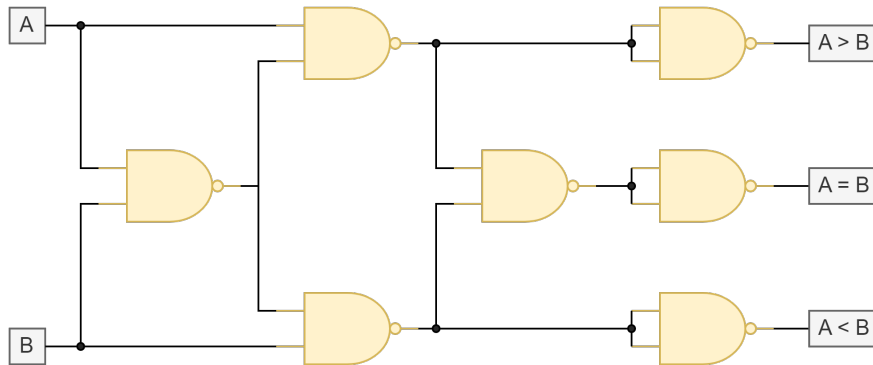
Figuur 11: Ontwerp van de logische schakeling van de 6-bit FULL-ADDER.

9 Vergelijking van binaire getallen

In de vierde opdracht zijn de volgende deelvragen onderzocht en beantwoord over vergelijking van binaire getallen:

1. Ontwerp een schakeling waarmee twee bits kunnen worden getest op ongelijkheid. Deze schakeling mag alleen maar bestaan uit NAND's;
2. Bouw de logische schakeling op het digiboard;
3. Ga de werking na en stel de bijbehorende waarheidstabel op.

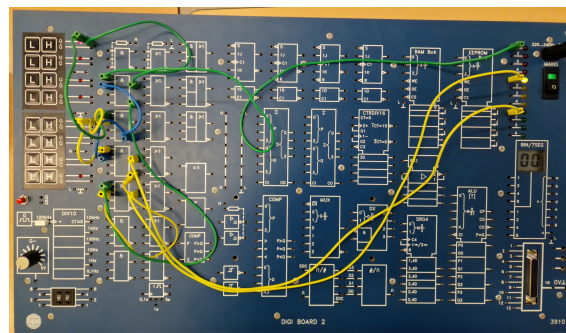
9.1 Ontwerp een schakeling waarmee twee bits kunnen worden getest op ongelijkheid. Deze schakeling mag alleen maar bestaan uit NAND's



Figuur 12: Ontwerp van de logische schakeling die de ongelijkheid/gelijkheid kan checken tussen twee bits.

9.2 Bouw de logische schakeling op het digiboard

Zie de foto hieronder:



Figuur 13: Foto van de gebouwde logische schakeling (comparator).

9.3 Ga de werking na en stel de bijbehorende waarheidstabel op

De werking van deze vergelijker (ook wel comparator genoemd) is als volgt:

1. $A \cdot \bar{B}$ checkt of $A > B$;
2. $(A \cdot B) + (\bar{A} \cdot \bar{B})$ (= XNOR-poort) checkt of $A = B$;
3. $B \cdot \bar{A}$ checkt of $A < B$.

Het is effectiever om de Booleaanse expressies van de NAND-poorten allemaal te vereenvoudigen, zodat de waarheidstabel dan makkelijker af te lezen is. Dit is mogelijk door de algemene wetten van de Booleaanse algebra toe te passen:

- $\overline{A \cdot B}$ wordt $\bar{A} + \bar{B}$;
- $\overline{A \cdot (\bar{A} \cdot \bar{B})}$ wordt $\bar{A} + (A \cdot B)$ wordt $B + \bar{A}$;
- $\overline{B \cdot (\bar{A} \cdot \bar{B})}$ wordt $\bar{B} + (A \cdot B)$ wordt $A + \bar{B}$;
- $\overline{(A \cdot (\bar{A} \cdot \bar{B})) \cdot (B \cdot (\bar{A} \cdot \bar{B}))}$ wordt $(A \cdot (\bar{A} + \bar{B})) + (B \cdot (\bar{A} + \bar{B}))$ wordt $(A \cdot \bar{B}) + (B \cdot \bar{A})$;
- $\overline{\overline{(A \cdot (\bar{A} \cdot \bar{B})) \cdot (B \cdot (\bar{A} \cdot \bar{B}))}}$ wordt $(\bar{A} + (A \cdot B)) \cdot (\bar{B} + (A \cdot B))$ wordt $(A \cdot B) + (\bar{A} \cdot \bar{B})$;
- $\overline{\overline{A \cdot (\bar{A} \cdot \bar{B})}}$ wordt $A \cdot (\bar{A} + \bar{B})$ wordt $A \cdot \bar{B}$;
- $\overline{\overline{B \cdot (\bar{A} \cdot \bar{B})}}$ wordt $B \cdot (\bar{A} + \bar{B})$ wordt $B \cdot \bar{A}$.

Zie de vereenvoudigde waarheidstabel hieronder:

A	B	$\bar{A} + \bar{B}$	$B + \bar{A}$	$A + \bar{B}$	$(A \cdot \bar{B}) + (B \cdot \bar{A})$	$(A \cdot B) + (\bar{A} \cdot \bar{B})$	$A \cdot \bar{B}$	$B \cdot \bar{A}$
1	1	0	1	1	0	1	0	0
1	0	1	0	1	1	0	1	0
0	1	1	1	0	1	0	0	1
0	0	1	1	1	0	1	0	0

10 Combinatorische logische schakeling

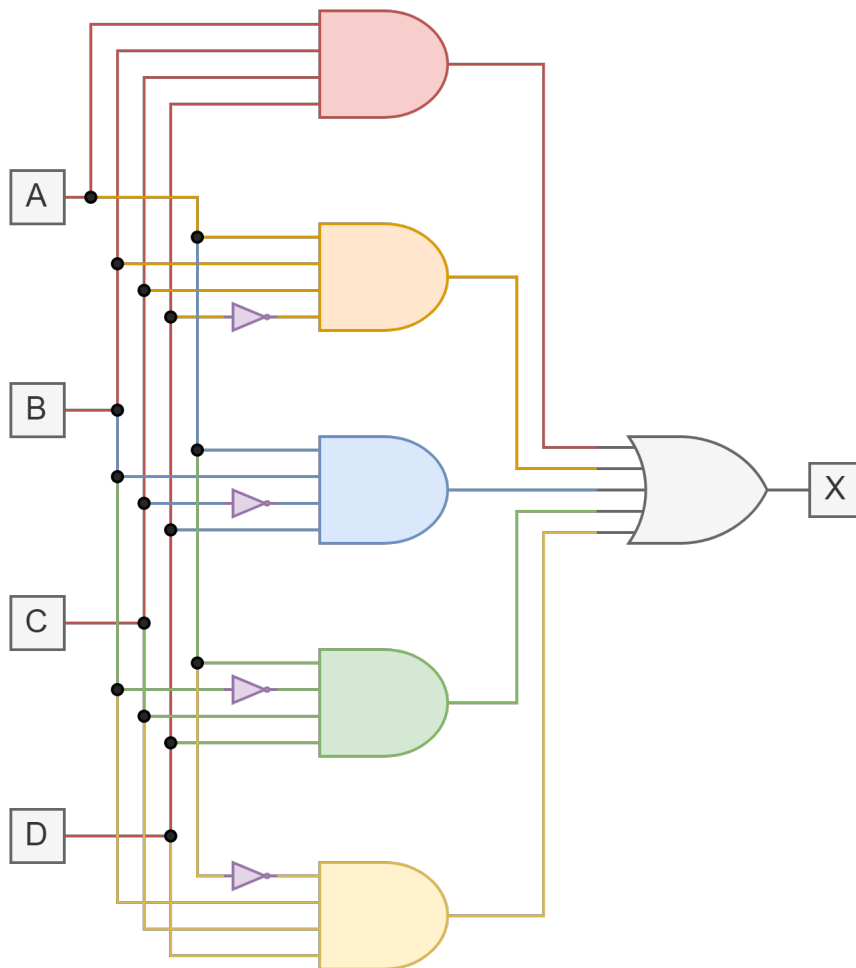
In de vijfde opdracht zijn de volgende deelvragen onderzocht en beantwoord over een combinatorische logische schakeling:

1. Bepaal de bij deze schakeling behorende SOP-uitdrukking;
2. Ontwerp de bijbehorende logische schakeling;
3. Bouw de logische schakeling op het digiboard.
4. Controleer de werking van deze schakeling. Maak voor de niveausensors gebruik van de schakelaars op het digiboard.

10.1 Bepaal de bij deze schakeling behorende SOP-uitdrukking

De SOP-uitdrukking is: $ABCD + ABC\bar{D} + AB\bar{C}D + A\bar{B}CD + \bar{A}BCD$

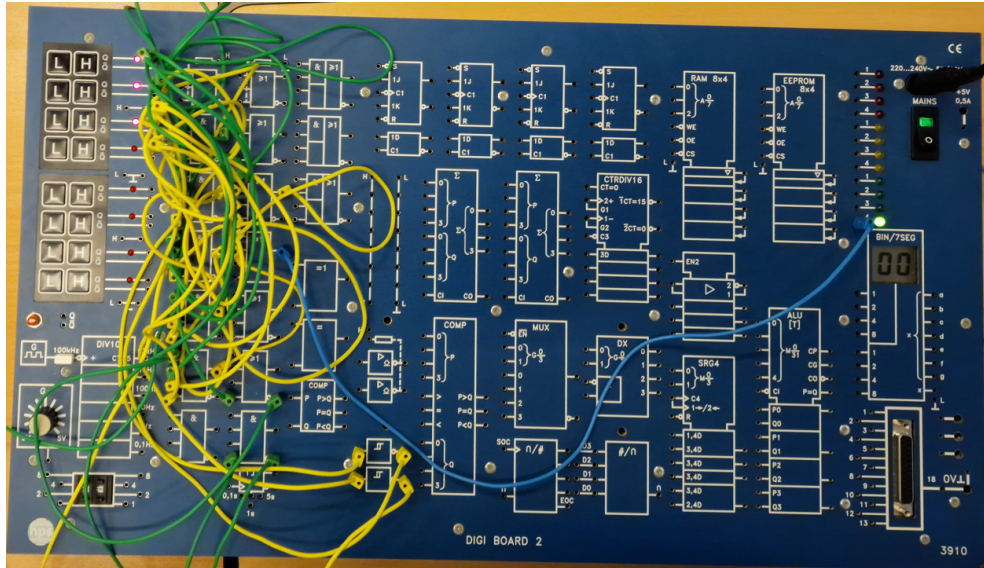
10.2 Ontwerp de bijbehorende logische schakeling



Figuur 14: Ontwerp van de logische schakeling van het tank-alarm systeem.

10.3 Bouw de logische schakeling op het digiboard

Zie de foto hieronder:



Figuur 15: Foto van de gebouwde logische schakeling van het tank-alarm systeem tijdens het practicum.

10.4 Controleer de werking van deze schakeling. Maak voor de niveausensors gebruik van de schakelaars op het digiboard

De schakeling werkt als volgt: als er minimaal drie van de vier tanks leeg zijn, dan gaat er een alarm af. De schakeling bevat vijf AND-poorten en één OR-poort. Er zijn vier tanks, dus dan zijn er vier mogelijkheden. Daarbij komt ook nog de vijfde AND-poort bij en die is bedoeld te signaleren indien alle tanks leeg zijn. De OR-poort zorgt ervoor dat als minimaal een van de AND-poorten actief hoog wordt, dat de OR-poort ook actief hoog wordt. Voor de niveausensors hebben we de groene LED gebruikt die op het digiboard zit. Als deze LED aan gaat betekent het dat drie van de vier tanks leeg zijn. Als alternatief voor de schakelaars is gebruik gemaakt van de HIGH/LOW knopjes die dienen als inputs van het digiboard, zie de foto 15.

11 Conclusie

In dit verslag is er onderzoek gedaan naar de werking van logische schakelingen, zoals een FULL-ADDER of een vergelijker. Daarbij hadden we ook een probleemstelling geïntroduceerd in H2 en de bijbehorende doelstellingen, namelijk:

- *“Hoe kunnen we op een effectieve manier diverse logische schakelingen ontwerpen en bouwen door het gebruik van zowel eenvoudige als geavanceerde logische poorten?”*

De bijbehorende doelstellingen:

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.

Bij de eerste en de tweede opdracht moesten we een HALF-ADDER en een FULL-ADDER bouwen door gebruik van logische poorten. We hebben de verschillen onderzocht tussen het gebruik van alleen basispoorten en geavanceerde logische poorten, zoals de XOR-poort. Daarnaast hebben we ook de werking van deze logische schakelingen onderzocht door de bijbehorende logische schakelingen te ontwerpen en de waarheidstabellen te maken.

Daarna hebben we in de derde opdracht verder moeten denken over hoe we met een 4-BIT FULL-ADDER ervoor kunnen zorgen dat we ook 6-BITS kunnen optellen. Dit probleem is opgelost door twee van deze FULL-ADDER's in te zetten.

In de vierde opdracht hebben we onderzocht hoe we binaire getallen met elkaar kunnen vergelijken. Deze vergelijker mocht alleen bestaan uit logische NAND-poorten en moet input A en B minimaal kunnen testen op ongelijkheid.

Tenslotte is er een combinatorische logische schakeling gebouwd die gebaseerd is op een alarm-systeem. In deze opdracht werden vier vloeistoftanks geïntroduceerd. Indien minimaal drie van de vier tanks leeg raakte, moest het alarm-systeem afgaan. Dit probleem is opgelost door vijf AND-poorten en één OR-poort te gebruiken op een logische wijze.

De bovenstaande probleemstelling is dan ook op een succesvolle manier beantwoord, namelijk door:

1. Het toepassen van de wetten van de Booleaanse algebra: door de wetten van de Booleaanse algebra toe te passen kun je geavanceerde Booleaanse expressies vereenvoudigen tot simpele Booleaanse expressies, dit leidt tot een effectieve wijze van het bouwen van logische schakelingen;
2. Het gebruiken van waarheidstabellen: door gebruik te maken van waarheidstabellen kun je gemakkelijk alle mogelijke toestanden overzichtelijk terug zien. Dit helpt bij het begrijpen van de logische functies- en schakelingen;
3. Het ontwerpen van logische schakelingen: door het doelgericht ontwerpen van een logische schakeling met daarbij behulp van een waarheidstabel kun je ook gemakkelijk ingewikkeldere logische schakelingen op een effectieve manier vereenvoudigen tot simpelere logische schakelingen.

We kunnen concluderen dat de vier doelstellingen van dit practicumverslag succesvol zijn behaald!

Kortom heeft dit practicum ervoor gezorgd dat we een sterke basis hebben gekregen over hoe diverse combinatorische logische schakelingen precies werken en hoe je deze op een effectieve manier kan toepassen.

12 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] —, “DIGI BOARD 2 (Type 3530) Datasheet,” <https://hps-systemtechnik.com/wp-content/uploads/content/english/3530.pdf>, 2014, accessed on [29-4-2023].
- [3] “Logische basispoorten,” Website, accessed: [30-4-2023]. [Online]. Available: <https://electricalengineering123.com/logic-gates-xor-nand-nor-and-gate-truth-table/>
- [4] 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>
- [5] iLogic, “3.2.5 truth tables for arguments,” bit.ly/3Vr0aSf, 2008, accessed: [2-5-2023].
- [6] “Booleaanse wetten,” Website, accessed: [7-5-2023]. [Online]. Available: <https://www.mi.mun.ca/users/cchaulk/misc/boolean.htm>
- [7] “Hps systemtechnik,” Website, accessed: [29-4-2023]. [Online]. Available: <https://hps-systemtechnik.com>
- [8] I. M. Copi, C. Cohen, and K. McMahon, *Introduction to Logic*. Routledge, Sep. 2016. [Online]. Available: <https://doi.org/10.4324/9781315510897>
- [9] “74xx series families,” Website, accessed: [30-4-2023]. [Online]. Available: <https://electronicsclub.info/74series.htm>
- [10] “Ic 7410,” Website, accessed: [30-4-2023]. [Online]. Available: <https://www.robotchbd.com/product/electronics-components/ic/analog/driver/ic7410/>
- [11] “Ic 7420,” Website, accessed: [30-4-2023]. [Online]. Available: <https://www.igelectronics.com/products/i88d0848bf/172681000000230282>
- [12] “Ic 7404,” Website, accessed: [30-4-2023]. [Online]. Available: <https://potentiallabs.com/cart/buy-7404-not-gate-ic-online-hyderabad-india>
- [13] SGS-THOMSON, “7404 Datasheet,” <https://datasheetspdf.com/datasheet/7404.html>, 1992, accessed: [30-4-2023].

- [14] FAIRCHILD, “7420 Datasheet,” <https://datasheetspdf.com/pdf-file/505339/FairchildSemiconductor/7420/1>, 1998, accessed: [30-4-2023].
- [15] PHILLIPS, “7410 Datasheet,” <https://datasheetspdf.com/pdf-file/854803/Philips/7410/1>, 1990, accessed: [30-4-2023].