Labo 4

# Materialen

We werken met het IDL800 experimenteertoestel en een oscilloscoop en gebruiken de volgende

componenten:

• SN74HC193: 4-bit synchrone, reversibele, up/down binaire counter

• 74HC93: 4-bit binaire ripple counter

Bestudeer grondig de datasheets van deze componenten. Ook andere standaard poorten (naar

eigen keuze) zijn nodig.

Als je een kloksignaal gebruikt, controleer daar ook de frequentie e.d. van op de scoop!

# Opdracht: Ripple counter deel A

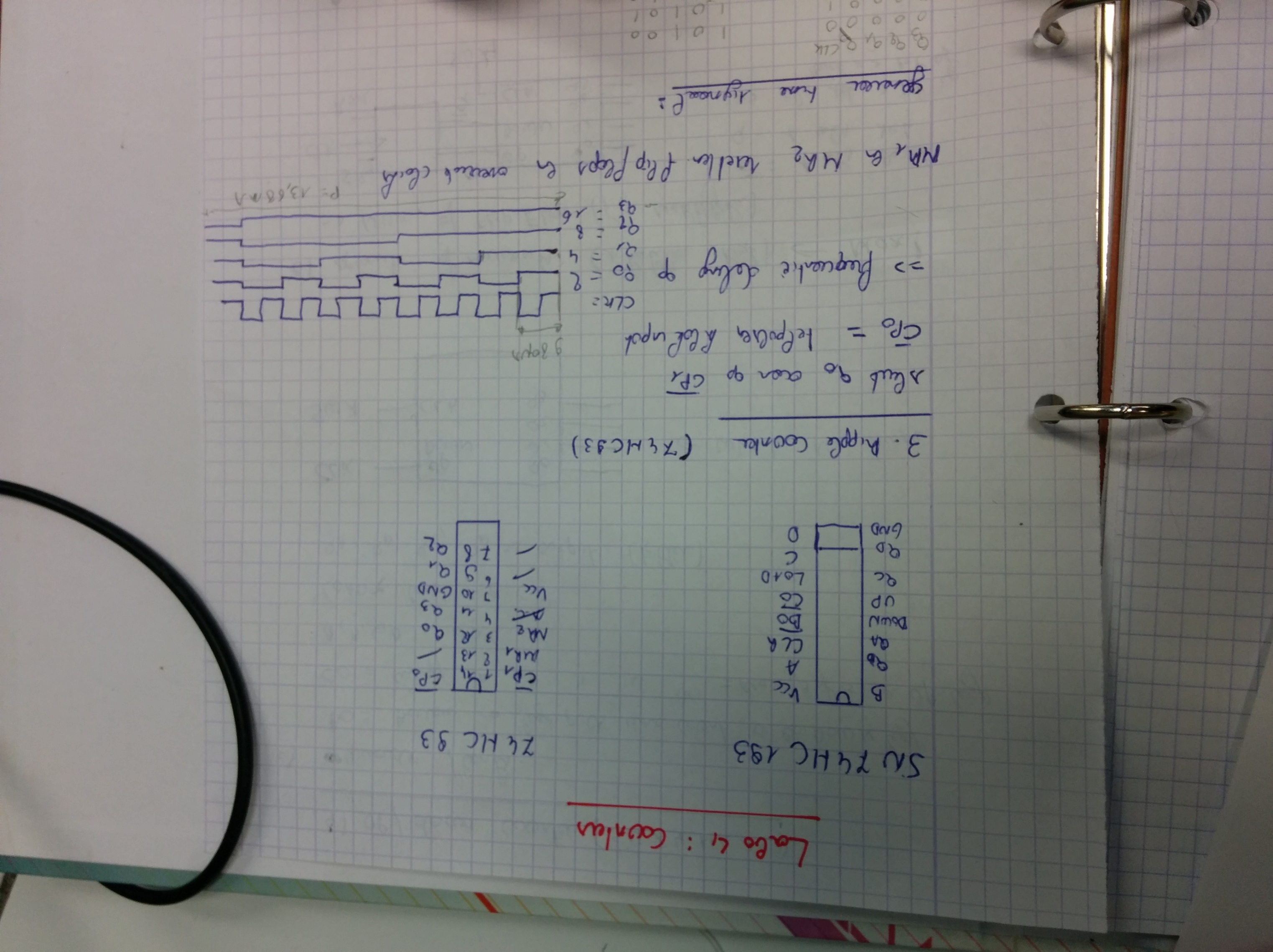
Tellerwerking demonstreren.

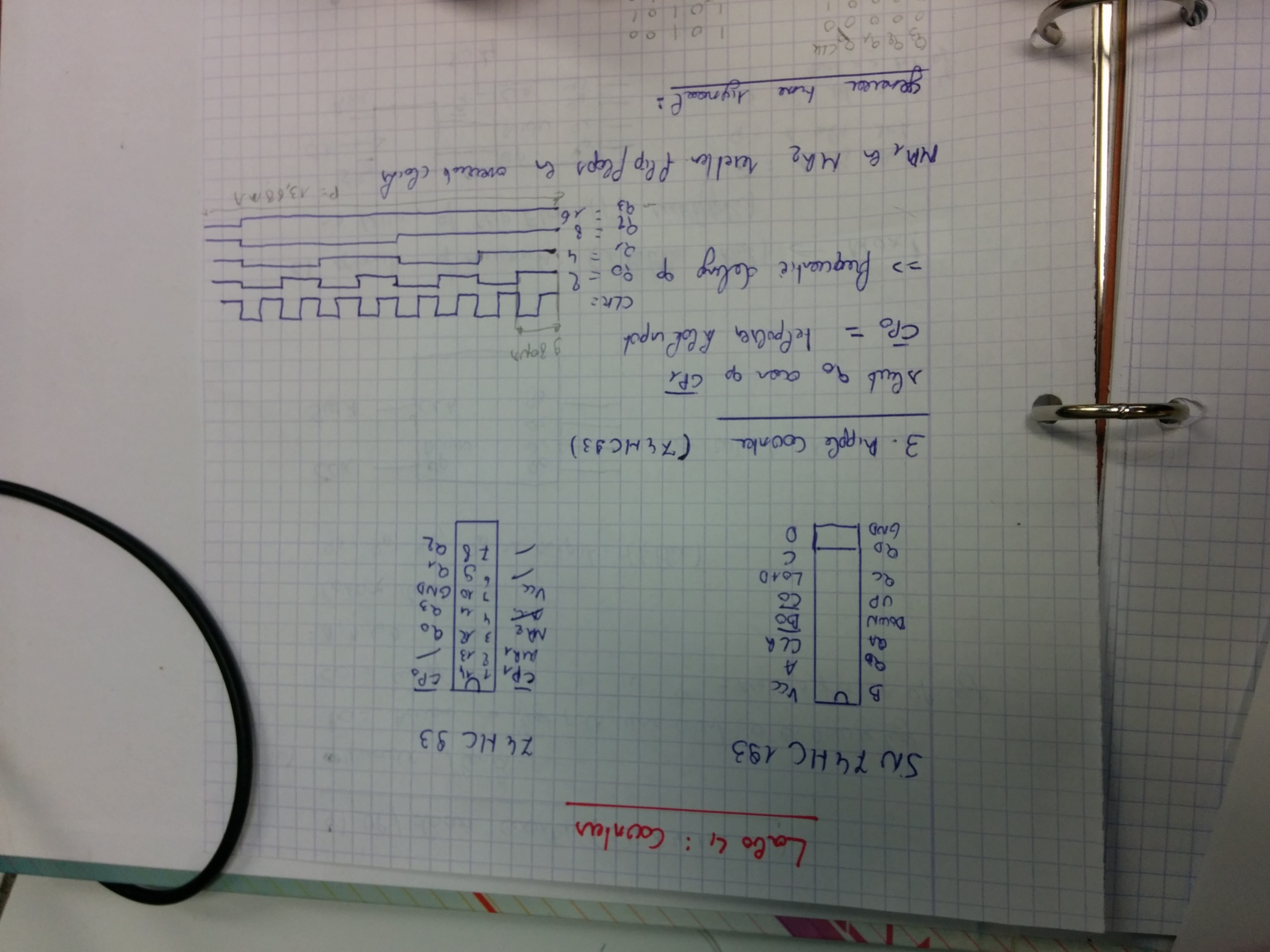
Leer de component kennen door hem eerst te doen tellen. (Vergeet niet Q0 met !CP1

te verbinden.)

Deze component gaat onze klok opdelen in verschillende tijdsignalen van 2 tot een macht.

## Pin layout





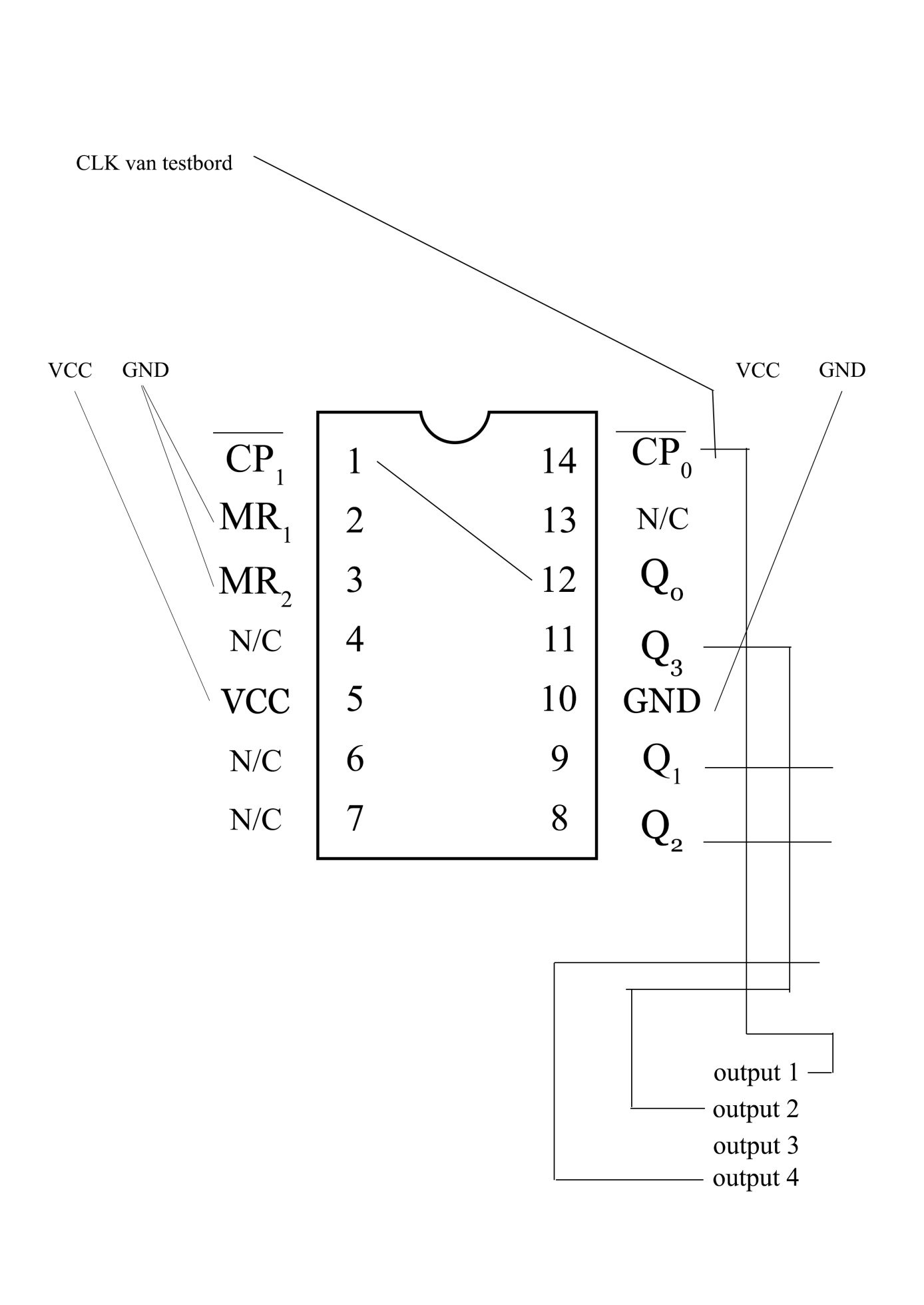
## Detail schakeling

Waarbij CP0 de klok van het testbord ontvangt.

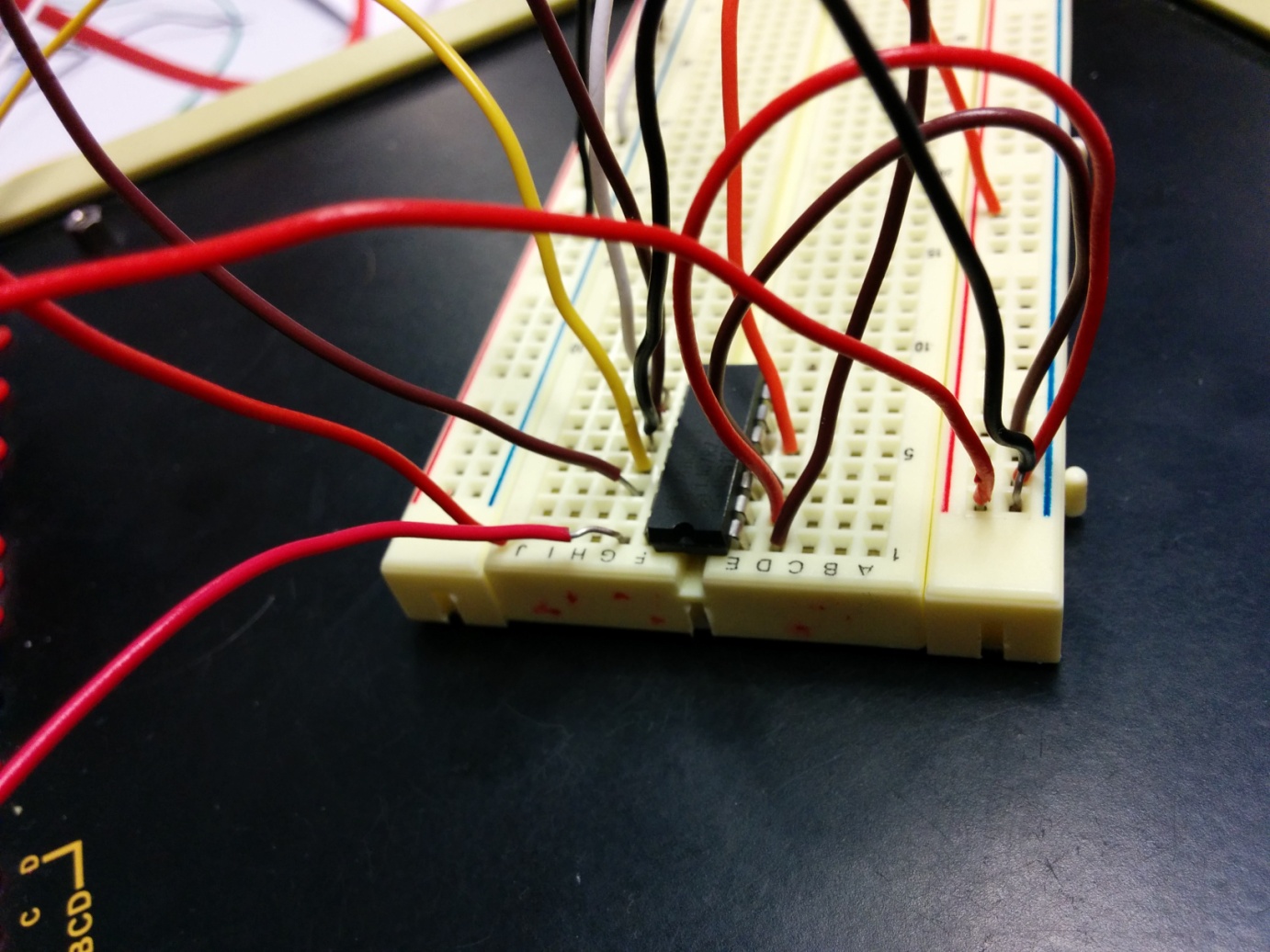
We zien op de oscilliscoop dat uitgang 1 onze originele klokpuls teruggeeft (die van ons testbord), en uitgang 2 geeft de klokpuls van QD (welke hoog gaat om de 8 klokpulsen van de ingangsklok, en dan terug laag na de volgende 8 pulsen) weer (wat de MSB is van de teller).

We kunnen dit dus samenvatten in volgende tabel:

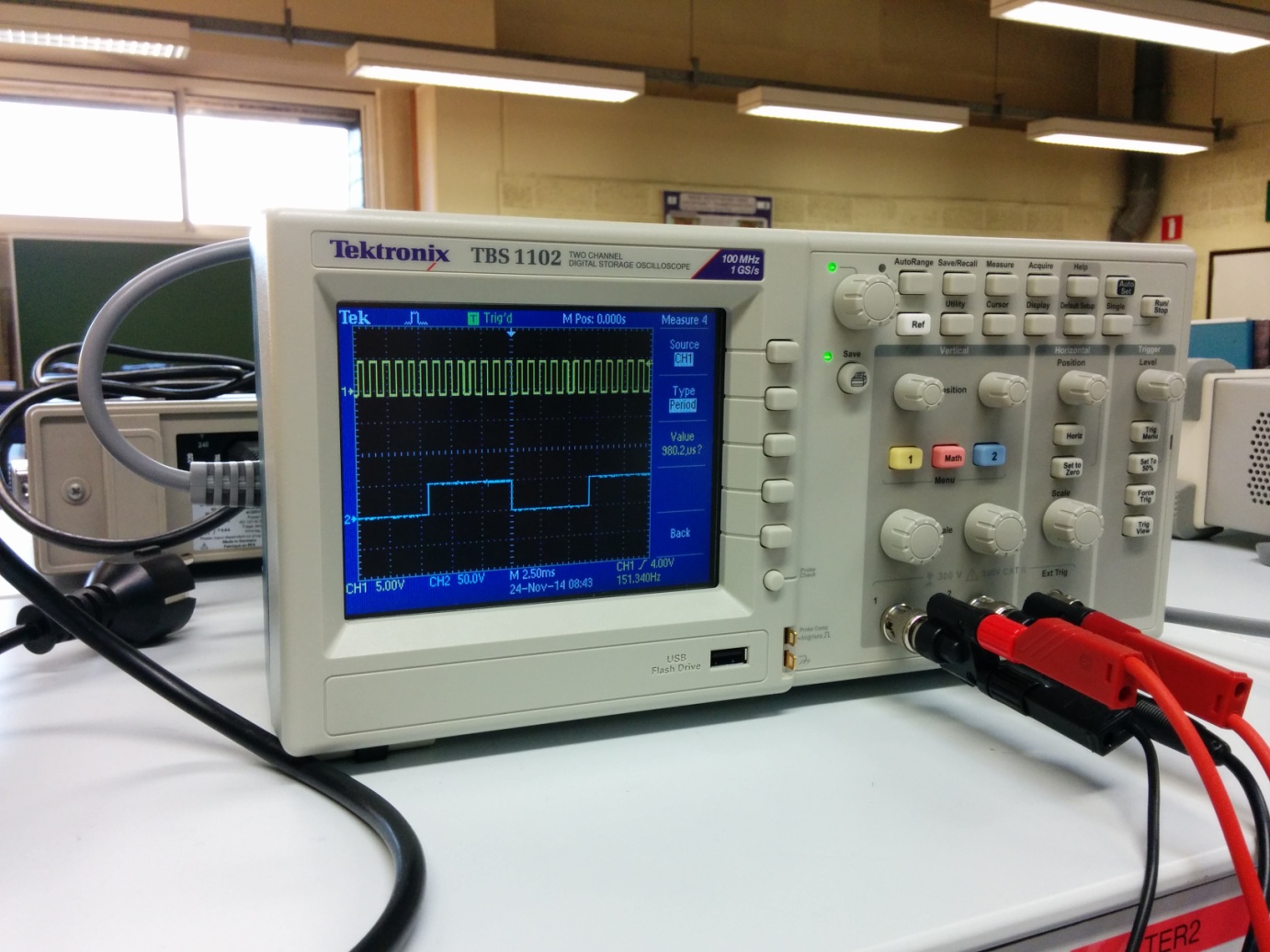
|  |  |  |  |
| --- | --- | --- | --- |
| QD | QC | QB | QA |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |



## Effectieve schakeling



## Resultaat



Waarbij de gele lijn (= originele klokpuls ) een periode heeft van 980 milliseconden

en de blauwe lijn ( = Q3 ) een periode heeft van 13,68 milliseconden.

# Opdracht: Ripple counter deel B

Bij deze opdracht moeten we het volgende signaal genereren: 1,1,1,1,0,0,0,0,0,0,1,1,0,0,0,0

We hebben dit niet kunnen testen in het labo, maar we hebben hier wel een theoretische benadering van kunnen doen.

Voor de oplossing hiervan weten we dat we een teller hebben die van 0 🡪 15 telt. Wat gelijk staat aan 16 verschillende staten.

Voor het patroon dat we gekregen hebben moeten we ook 16 staten hebben, wat dus perfect mogelijk is met deze counter.

Voor de oplossing hebben we gekozen om een waarheidstabel te maken die de output geeft die we nodig hebben en dit voor respectievelijk staten 0 tot en met 16.

Daarna gaan we deze tabel omzetten in een schakeling (wat wij aan de hand van een Karnaugh Kaart gedaan hebben).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| State | Q3 | Q2 | Q1 | Q0 | Qout |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 0 |
| 6 | 0 | 1 | 1 | 0 | 0 |
| 7 | 0 | 1 | 1 | 1 | 0 |
| 8 | 1 | 0 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 0 |
| 10 | 1 | 0 | 1 | 0 | 1 |
| 11 | 1 | 0 | 1 | 1 | 1 |
| 12 | 1 | 1 | 0 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 | 0 |
| 14 | 1 | 1 | 1 | 0 | 0 |
| 15 | 1 | 1 | 1 | 1 | 0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | **Q3** | **Q2** |  |
|  |  | 00 | 01 | 11 | 10 |
| **Q1** | 00 | 1 | 0 | 0 | 0 |
|  | 01 | 1 | 0 | 0 | 0 |
| **Q0** | 11 | 1 | 0 | 0 | 1 |
|  | 10 | 1 | 0 | 0 | 1 |

## Waarheidstabel Karnaugh kaart

🡺 Bij een Karnaugh Kaart nemen we zo groot mogelijke groepen welke we dan omzetten in Booleaanse Algebra.

Dit geeft ons als oplossing: N(Q3Q2) + Q0

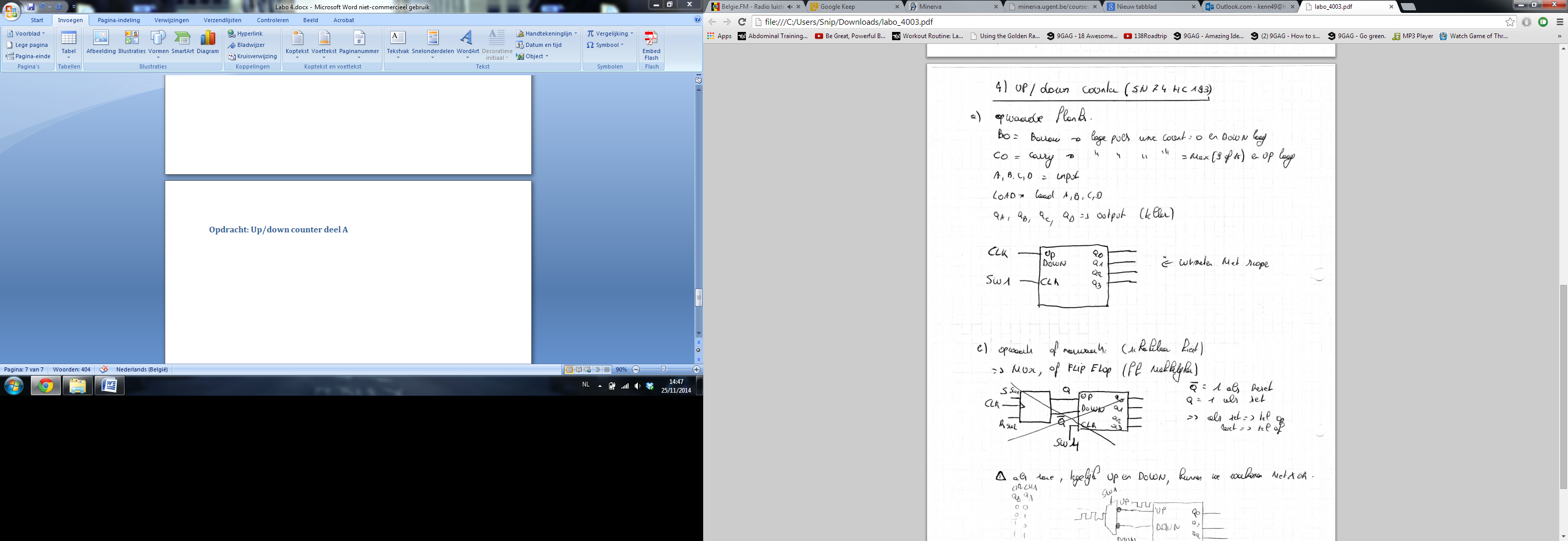
Wat we vereenvoudigen naar: N(Q3) + N(Q2) + Q0 (wetten van de morgan)

## C:\Users\Snip\Downloads\media-20141124 (6).jpgSchakeling

Vervolgens kunnen we de schakeling opstellen

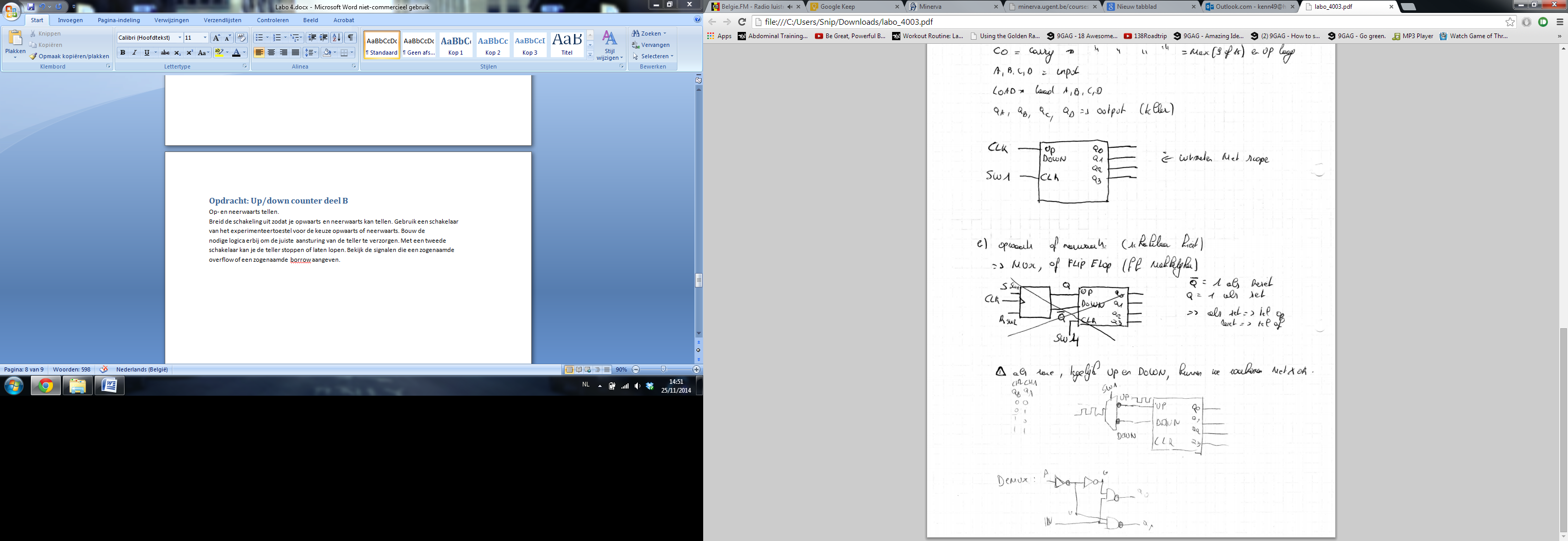
# Opdracht: Up/down counter deel A

We hebben deze opdracht voorbereidt maar niet uitgevoerd in de praktijk, dit omdat we vonden dat deel B van deze opdracht interessanter was en we dit beter konden doen met de resterende tijd.



# Opdracht: Up/down counter deel B

We hadden de keuze tussen een Set-Reset flipflop of een Multiplexer. We hebben hier gekozen voor de Multiplexer omdat we dit beter vonden passen binnen onze opdracht. Ook gaf ons dit meer variatie voor eens een andere component aan te raken in de praktijk. Een flipflop had hoogst waarschijnlijk een goedkopere oplossing geweest doordat deze minder componenten gebruikt.

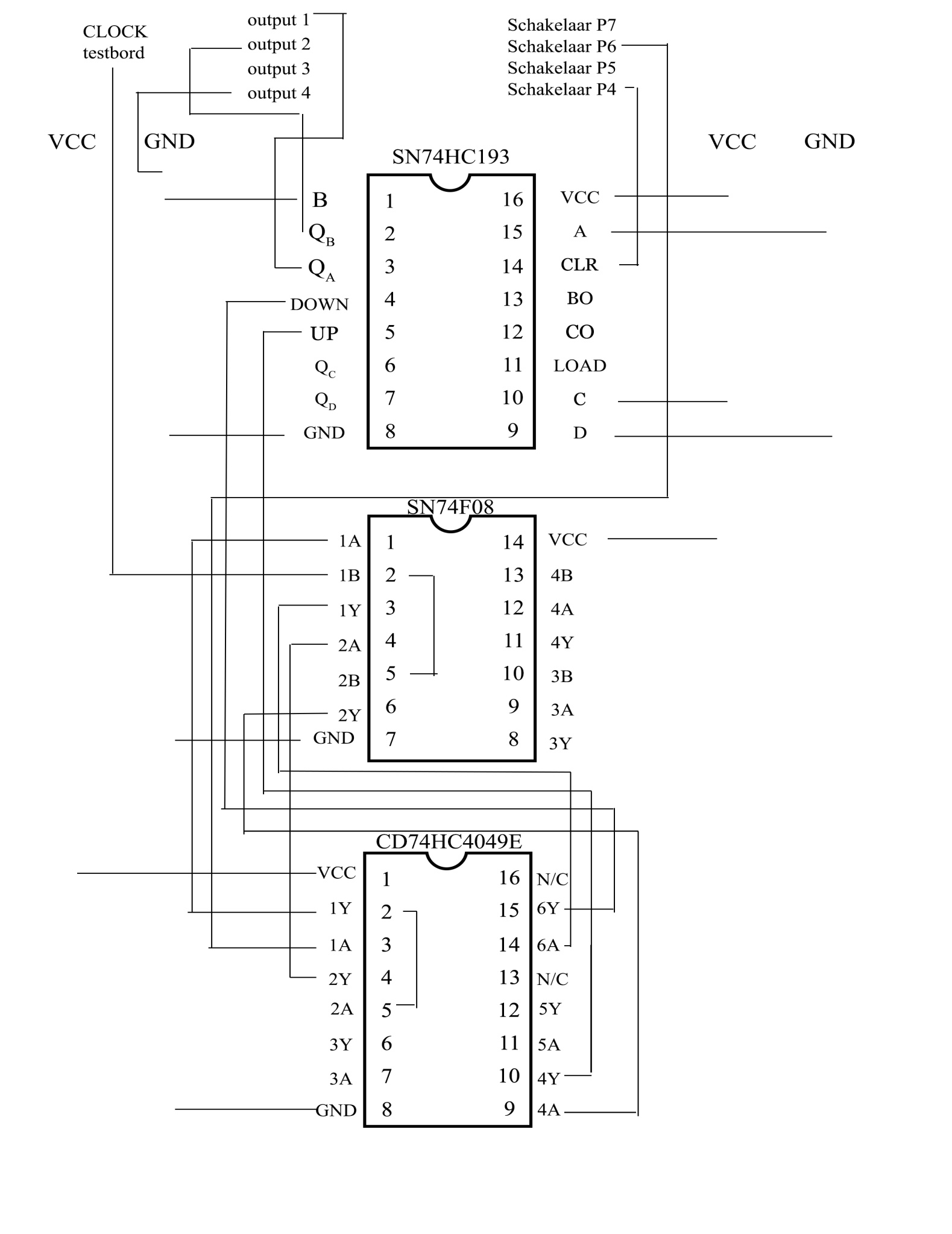


## Detail schakeling

eerste chip is de op en neer teller.

tweede chip is and-chip.

derde chip is not-chip.



# Opdracht: Up/down counter deel C

Hebben we niet gedaan tijdens het Labo