

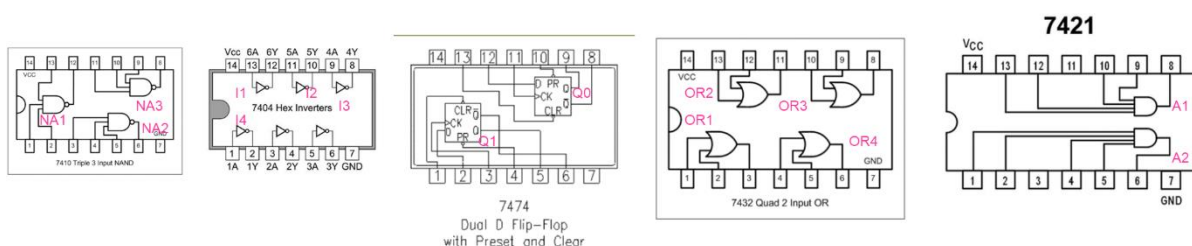
# Stappenplan voor het maken van een synchrone logische schakeling

1. Ontwerp een logische schakeling met de Quick and Dirty of de Compact and Clean methode (gebruik voor de eindopdracht van DIT Compact and Clean)
2. Bouw en test de schakeling in Logisim.
3. Kijk op het document "Onderdelen-DIT" op canvas welke chips met logische poorten en flipflops je ter beschikking hebt.
4. Kopieer je Logisim-schakeling, en vervang de gates die je gebruikt door gates die je ter beschikking hebt. Gebruik daarvoor eventueel de regel van de Morgan. Test in Logisim of het nog steeds goed werkt.
5. Print het circuit uit. Geef elke poort een unieke naam (bijvoorbeeld A1 voor de eerste AND poort, A2 voor de tweede, O1 voor de eerste or-poort, etc).
6. Prik de chips met alle gates en flipflops die je nodig hebt op een of meerdere breadboards.
7. Maak met behulp van de chip-layouts in "Onderdelen-DIT" (of de datasheets) een tekening van de chips zoals ze op het breadboard zitten (zie figuur 1 verderop voor een indruk). Je kunt het jezelf makkelijk maken door de plaatjes uit "Onderdelen-DIT" te copy-pasten naar een tekenprogramma (zoals draw.io).
8. Bedraad je breadboard. Daarbij wordt bijna altijd wel een foutje gemaakt. Die kun je systematisch opsporen:
9. Hang aan de uitgang (Q) van elke flipflop een led (via voorschakelweerstand) naar ground, zodat je te allen tijde kunt zien in welke toestand je zit.
10. Doe hetzelfde met de ingang (D) van elke flipflop, zodat je ten allen tijde kunt zien naar welke toestand hij wil gaan bij de volgende klokpuls.
11. Maak met de Arduino-IDE een betrouwbare, handgestuurde klokpuls (zie figuur 2 verderop): een druk op de enter-toets genereert dan een enkele klokpuls: klok-hoog, klok-laag. (als je dat met een gewone drukknop probeert, heb je te veel last van fysiek "gedender", en krijg je vaak meer dan 1 opgaande flank per indrukken)
12. Bouw het handige meetcircuit (zie figuur 3 verderop). Je kunt dan met een lange breadboard-draad de spanningen in je circuit inspecteren, en controleren of ze overeenkomen met de spanningen in logisim:  
Als beide leds branden zweeft het knooppunt. Dan is er zowiezo iets mis.  
Alleen een brandende groene led betekent een logische 1 (5V). Alleen een brandende rode led betekent logische 0 (0V).
13. Vergeet niet van elke chip de voeding en ground pinnen aan te sluiten.
14. Je flipflops hebben ook "NOT Preset" en "NOT Clear" input pinnen. Vergeet niet om die ook aan te sluiten. Als je ze niet gebruikt, hang ze dan aan de +5V.
15. Als je maar een deel van de inputs van een gate gebruikt, sluit de overige inputs dan aan op +5V of ground: Als je van bijvoorbeeld een 4-input AND maar 3 inputs gebruikt, sluit de 4<sup>e</sup> input dan aan op +5V.
16. Zorg dat elke button een pulldown of pullup weerstand heeft, zodat zijn output signaal nooit zweeft: Een "positieve logica" button zet je tussen +5V en het signaal wat je wil maken (laten

we zeggen signaal B1 (van Button1). Verbindt dan B1 met een pull-down weerstand (van bijvoorbeeld 2kOhm) naar ground. Een “negatieve logica” button (die je bijvoorbeeld kunt gebruiken om de (NOT-Clear) inputs tijdelijk naar ground te trekken) maak je andersom: zet zo’n button tussen ground en signaal B1 en verbindt B1 via een pull-up weerstand naar +5V.

#### 17. Foutzoek-procedure:

- Bepaal aan de hand van de flipflop-output leds in welke toestand je je bevindt.
- Bepaal aan de hand van je toestand-overgangsdiagram wat er moet gebeuren bij de volgende klokpuls.
- Maak de klokpuls (enter op je toetsenbord) (eventueel terwijl je een Button ingedrukt houdt) en controleer of het gewenste gedrag optreedt.
- Zo niet, zoek de rij op in de dubbel-Q karnaugh diagram die overeenkomt met de toestand waarin je je bevindt.
- Kijk in de kolom met de knop die je indrukte (of juist niet) wat daar ingevuld staat. Stel daar staat 10 ingevuld. Dat betekent dat Q1next (en dus de D input van flipflop1) op 1 moet staan en Q0next (en dus de D input van flipflop0) op 0.
- Controleer dat met het handige meetcircuit (figuur 3 verderop), gebruikmakend van je tekening van je breadboard (figuur 1 verderop).
- Als een van beide niet klopt, Laten we zeggen dat we op Q1next een 0 meten ipv een 1, kijk dan in je Logisim circuit-tekening welke gate die 0 levert.
- Controleer de inputs van die gate met het handige meetcircuit. Bepaal van de input die niet klopt weer van welke gate dat de output is. Herhaal deze stap totdat je de fout gevonden hebt. Je begint dus aan het eind (bij Q1next) en test zo terug naar voren. Die binary-search approach garandeert dat je de fout in weinig stappen vindt.



Figuur 1 Tekening van chips op breadboard met gate-annotaties

```
int incomingByte = 0; // for incoming serial data
int clockOutPin = 9;

void setup() {
  Serial.begin(9600); // opens serial port, sets data rate to 9600 bps
  pinMode(clockOutPin, OUTPUT);
}

void loop() {
  // send data only when you receive data:
  if (Serial.available() > 0) {
    // read the incoming byte:
```

```

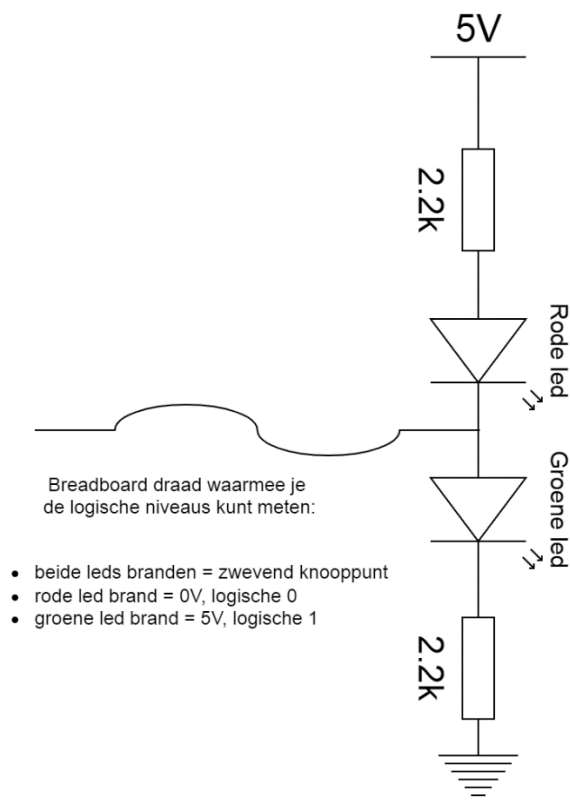
incomingByte = Serial.read();

// say what you got:
Serial.print("I received: ");
Serial.println(incomingByte, DEC);
digitalWrite(clockOutPin , HIGH);
delay(1); // 1ms delay - probably not needed.
digitalWrite(clockOutPin , LOW);
}
}

```

*Figuur 2 Arduino code voor het genereren van een klokpuls door de enter toets in te drukken na selectie van het input-veld van de serial monitor in de Arduino IDE*

## Handig meetcircuit voor het opsporen van fouten in een logische schakeling



*Figuur 3 Handig meetcircuit voor logische schakelingen*