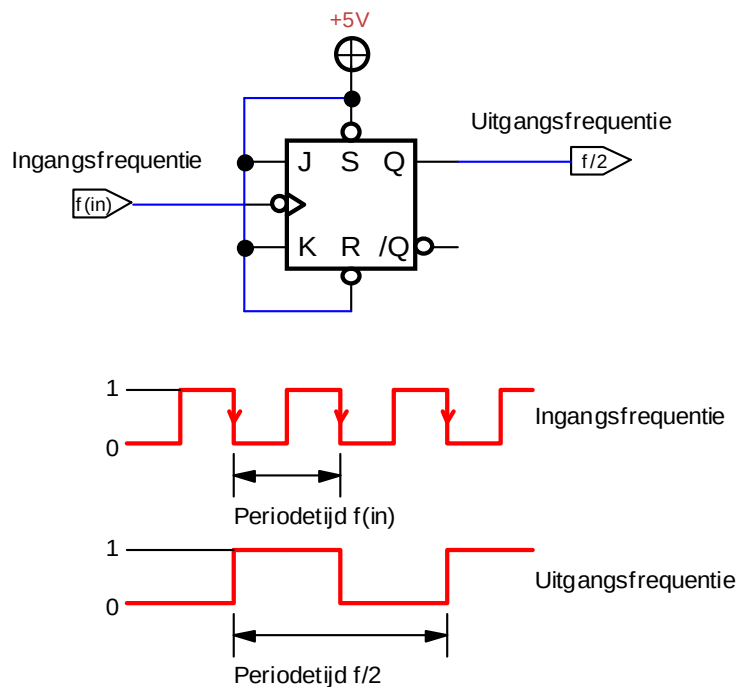


4.5 Asynchrone tellers

4.5.1 Principe van frequentiedeling

We kunnen op een eenvoudige manier een JK-flipflop doen werken als een zogenaamde **2-deler**. Hiertoe verbinden we J-ingang en de K-ingang aan het logisch '1' (H) niveau. Door dit opzet staat er op de Q-uitgang een bloksignaal waarvan de frequentie juist de *helft* is van de frequentie van het aangelegde kloksignaal.

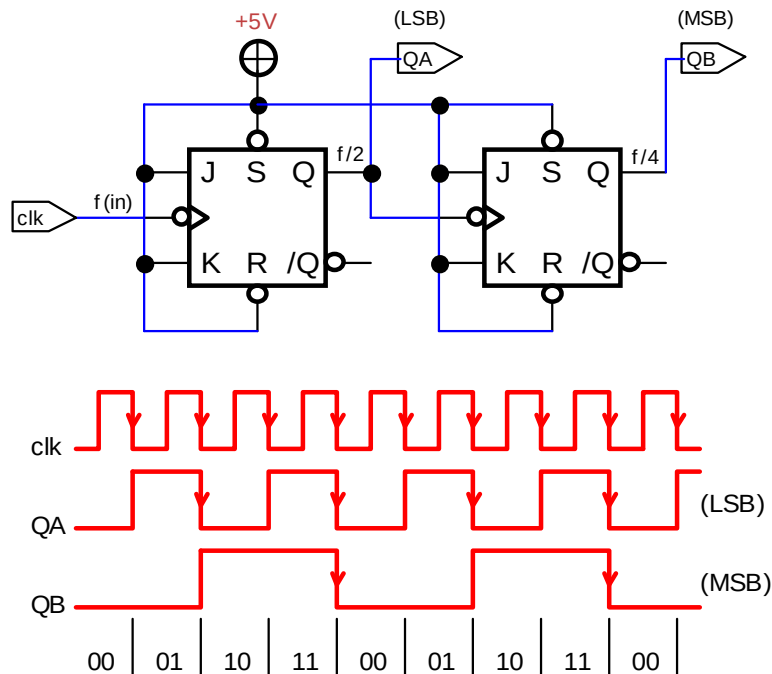


△ Fig.4.45 | Principe van een frequentiedeler.
Bron: W. Van Wichelen

Het circuit van figuur 4.45 gedraagt zich dus als een **frequentie-deler** die de ingangsfrequentie (Clk) juist deelt door twee (2)^[26]. Is de ingangsfrequentie bijvoorbeeld 2 kHz, dan bekomen we een uitgangsfrequentie van 1 kHz. Als we een tweede flipflop bij-schakelen, waarbij we de uitgang van de eerste flipflop door-verbinden met de klokingang van de tweede, dan resulteert dit in een frequentie die juist één *vierde* is van de oorspronkelijke klokfrequentie.

[26] Een frequentie halveren komt neer op een 'toonafstand' van juist 1 oktaaf.

Plaatsen we alle uitgangstoestanden van de frequentiedeler van figuur 4.46 na elkaar, dan bekomen we volgende sequentie: '00', '01', '10', '11',... De cascade van flipflops gedraagt zich als een **telschakeling** die van $(0)_{10}$ tot $(3)_{10}$ telt en genereert also vier verschillende *teltoestanden*.



△ Fig.4.46 | Een frequentiedeler met een deling door 4.
Bron: W. Van Wichelen

Principe

Algemeen resulteert een cascade van **n**-aantal flipflops in een **frequentiedeling** met factor 2^n .



Definitie

Telschakelingen waarbij de eerste flipflop getriggerd wordt door een kloksignaal en alle andere flipflops, die deel uitmaken van de cascade, worden getriggerd door de uitgang van de voorganger worden '**ripple counters**' genoemd.



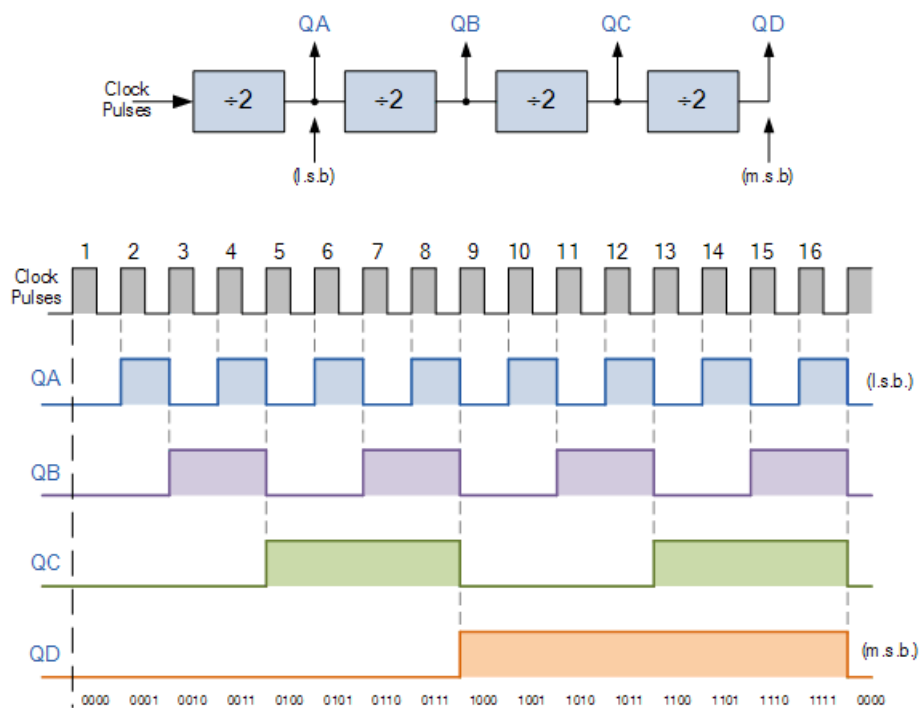
Omdat de uitgangstoestanden van de flipflops *niet* tegelijkertijd veranderen bij elke klokpuls spreken we van **asynchrone tellers**.

4.5.2 Binaire modulo-tellers

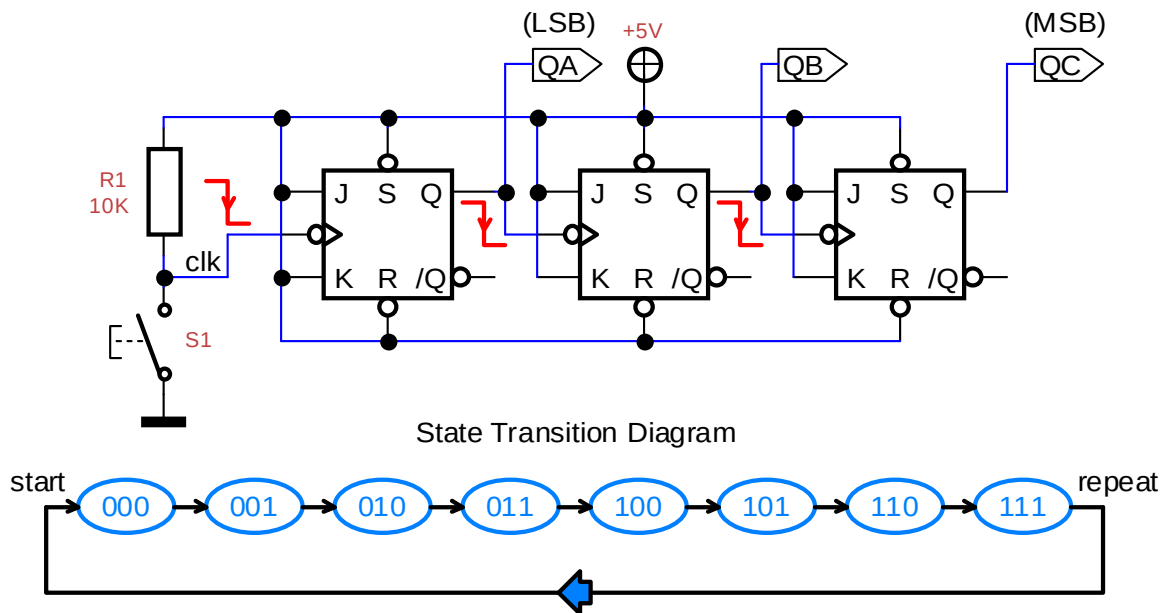
Om een asynchrone binaire teller te maken dienen we een n-aantal flipflops in cascade te plaatsen waarbij 'n' overeenkomt met het aantal mogelijke teltoestanden. We spreken dan kortweg over de **modulus** van een teller.

Enkele voorbeelden maken het concreet:

- 3-bit binaire teller bezit $2^3 = 8$ mogelijke teltoestanden
deze **modulo-8 teller** telt van $(0)_{10}$ t.e.m. $(7)_{10}$
- 4-bit binaire teller bezit $2^4 = 16$ mogelijke teltoestanden
deze **modulo-16 teller** telt van $(0)_{10}$ t.e.m. $(15)_{10}$
- 5-bit binaire teller bezit $2^5 = 32$ mogelijke teltoestanden
deze **modulo-32 teller** telt van $(0)_{10}$ t.e.m. $(31)_{10}$



△ Fig.4.47 | Tijddiagram van een modulo-16 ripple counter
Bron: <https://www.electronics-tutorials.ws>



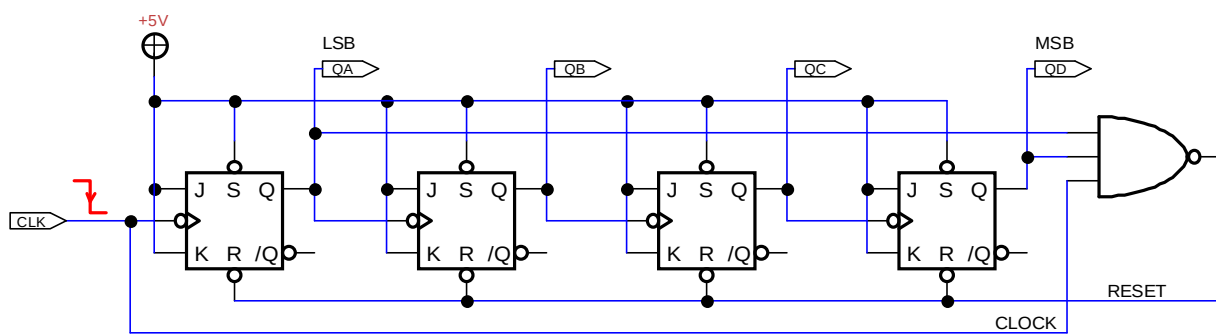
Clock Cycle	Output Bit Pattern		
	QC	QB	QA
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

△ Fig.4.48 Principe van een modulo-8 ripple counter
 Schema: W. Van Wichelen
 Bron: <https://www.electronics-tutorials.ws>

We kunnen nu modulo-tellers bouwen met een modulus die het resultaat is van een *macht van twee*. Soms kan het echter nodig zijn dat een teller zichzelf eerder reset in plaats van de natuurlijke opeenvolging verder te zetten. Wat dacht je bijvoorbeeld van een *modulo-5* teller of een *modulo-10* teller?

4.5.3 Asynchrone decadeteller

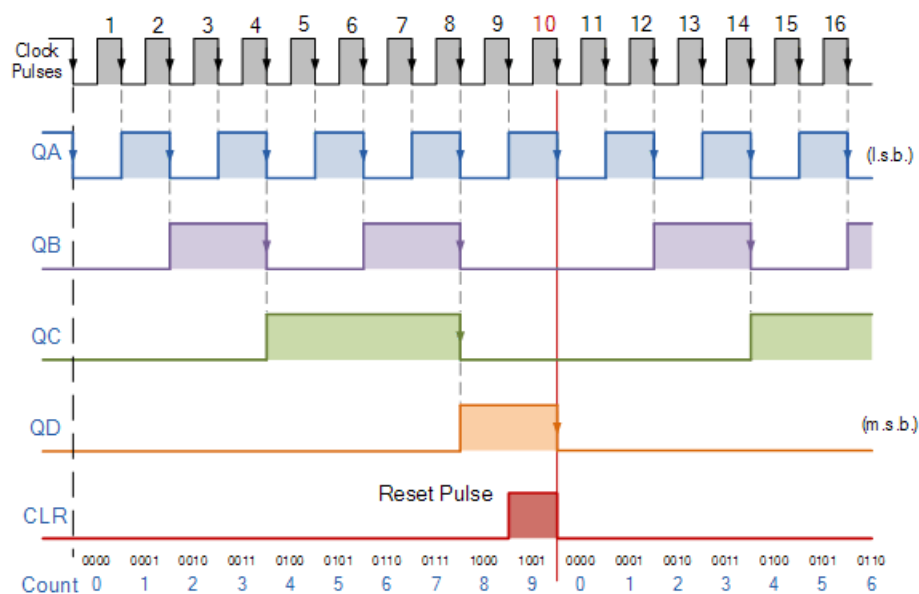
Laten we de asynchrone modulo-16 teller uitbreiden met wat bijkomende combinatorische logica zodat we een zogenaamde **decadeteller** bekomen. Een decadeteller bezit 10 teltoestanden en telt dus van $(0)_{10}$ t.e.m. $(9)_{10}$. Dit tellertype passen we veelvuldig toe bij standaard decimale rekenkundige bewerkingen. Een decadeteller moeten we terug op nul plaatsten wanneer hij de decimale waarde '10' bereikt. Deze voorwaarde $(1010)_2$ kunnen we gebruiken om via een NAND-poort alle flipflops, die deel uitmaken van de cascade, te resetten. Nadeel van deze methode is dat de teller heel eventjes (de vertragingstijd van de NAND-poort) een 11^e toestand $(10)_{10}$ genereert om daarna ogenblikkelijk te resetten naar $(0)_{10}$. Bij trage toepassingen, zoals een digitale klok, hoeven we hier gelukkig geen rekening mee te houden. Figuur 4.49 toont een alternatieve methode die gebruik maakt van JK-flipflops met actief lage resetingangen.



△ Fig.4.49 | Principe van een asynchrone decadeteller.
Bron: W.Van Wichelen

Bij elke flankovergang van het ingangskloksignaal telt deze teller opwaarts van $(0000)_2$ tot $(1001)_2$. Op het moment dat de teltoestand gelijk is aan $(9)_{10}$ zijn uitgang Q_A en uitgang Q_D beiden logisch '1'. Wanneer de volgende klokpuls zich aandient, zullen alle 3 de ingangen van de NAND-poort logisch '1' worden en zal de NAND-poort een logische '0' afleveren. De resetingangen van de toegepaste JK-flipflops zijn *actief laag*. Bijgevolg zullen bij de 10^e klokpulsovergang alle flipflops gereset worden. De twee bovenste ingangen van de NAND-poort zijn terug logisch '0' zodat de NAND nu een logisch '1' niveau op de RESET-lijn plaatst. De telcyclus kan weer van voor af aan beginnen. Figuur 4.50 toont de toestandentabel van de hier beschreven decadeteller.

Clock Count	Output bit Pattern				Decimal Value
	QD	QC	QB	QA	
1	0	0	0	0	0
2	0	0	0	1	1
3	0	0	1	0	2
4	0	0	1	1	3
5	0	1	0	0	4
6	0	1	0	1	5
7	0	1	1	0	6
8	0	1	1	1	7
9	1	0	0	0	8
10	1	0	0	1	9
11	Counter Resets its Outputs back to Zero				

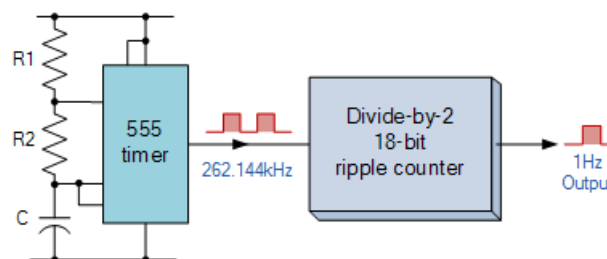


△ Fig.4.50 | Toestandentabel en tijddiagram van een decaden-teller. Bron: <https://www.electronics-tutorials.ws>

Door in het principeschema van figuur 4.49 simpelweg andere Q-aansluitingen voor de NAND-ingangen te kiezen of een andere logische poortcombinatie toe te passen, is elke tellermodulus mogelijk. Een modulo-12 teller kan gemakkelijk verwezenlijkt worden door Q_C en Q_D met de 2 bovenste ingangen van de NAND-poort te verbinden. Het binaire equivalent voor $(12)_{10}$ is immers $(1100)_2$ indien Q_A de minst betekende bit (LSB) voorstelt. Omdat de maximale modulus bepaald wordt door 2^n (waarbij 'n' staat voor het aantal flipflops) is het nu niet moeilijk om uit te maken hoeveel flipflops je nodig hebt om een willekeurige modulo-teller te realiseren: de laagste macht van twee moet immers groter of gelijk zijn aan de gewenste modulus.

4.5.4 Praktische 1Hz klok

Een typische toepassing van een asynchrone ripple-counter is om een hoge klokfrequentie te reduceren tot een handigere frequentie voor bijvoorbeeld de secondenteller in digitale klokken. Het is vrij eenvoudig om met een 555-timer een 1Hz blokgolf te genereren maar de stabiliteit en de nauwkeurigheid van deze lage frequentie is niet om over naar huis te schrijven! Algemeen mogen we er van uit gaan dat we een zo hoog mogelijke 555-timer frequentie moeten toepassen om de afwijkingen van het 1Hz signaal binnen redelijke grenzen te houden. We kunnen de 555-timer op de geëigende manier instellen als astabiele multivibrator met een uitgangsfrequentie van bijvoorbeeld 262,144kHz. Als we de 555-timer laten volgen door een ripple-counter die bestaat uit een cascade van **18** flipflops, dan genereren we een redelijk precies 1Hz timing-signaal.



△ Fig.4.51 | Principe van een 1Hz-klok
Bron: <https://www.electronics-tutorials.ws>

Wanneer het belangrijk is uiterst stabiele en nauwkeurige signaalfrequenties te produceren dan is een 555-timer af te raden. Beter is gebruik te maken van hoog frequente **kristaloscillatoren** in combinatie met multi-bit frequentiedelers.

4.5.5 Propagation delay

Het voornaamste kenmerk van een asynchrone teller is dat de flipflops niet aangestuurd worden door *eenzelfde* kloksignaal. De signaaltoestand van de uitgangen is immers afhankelijk van de voorliggende flipflop in de keten. Een belangrijk nadeel van asynchrone tellers is dan ook de kleine **tijdvertraging** die er is tussen het ingangskloksignaal en de verandering van de signaaltoestand van de uitgangen. Dit is te wijten aan de interne schematuur van de poortlogica. We spreken bij asynchrone ontwerpen van '**propagation delay**' die bij zeer hoge klokfrequenties kan leiden tot onstabiel gedrag van de tellers. Dit is dan ook de reden waarom je dit type teller niet mag toepassen bij hoogfrequente telcircuits en bij multibit ontwerpen. Algemeen mag je stellen dat voor elke flipflop die je toevoegt bij een asynchrone teller je inboet op de maximale toepasbare klokfrequentie. Naast asynchrone tellers bestaan er ook *synchrone* tellers en die kennen het probleem van 'propagation delay' gelukkig niet.

Laboproef 7

Realiseer een **asynchrone decadeteller** m.b.v. JK-flipflops die telt van $(0)_{10}$ t.e.m. $(9)_{10}$.

Doe eerst een computersimulatie met afzonderlijke JK-flipflops vooraleer je de schakelingen in praktijk brengt op een breadboard.

Maak gebruik van LED's om de respectievelijke toestanden van de uitgangen aan te geven.

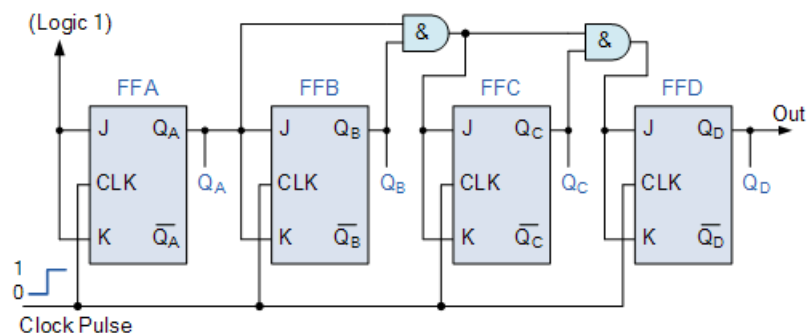
Pas een klokfrequentie toe van ongeveer 1 hertz m.b.v. een 555-timer.

4.6 Synchronische tellers

4.6.1 Principe

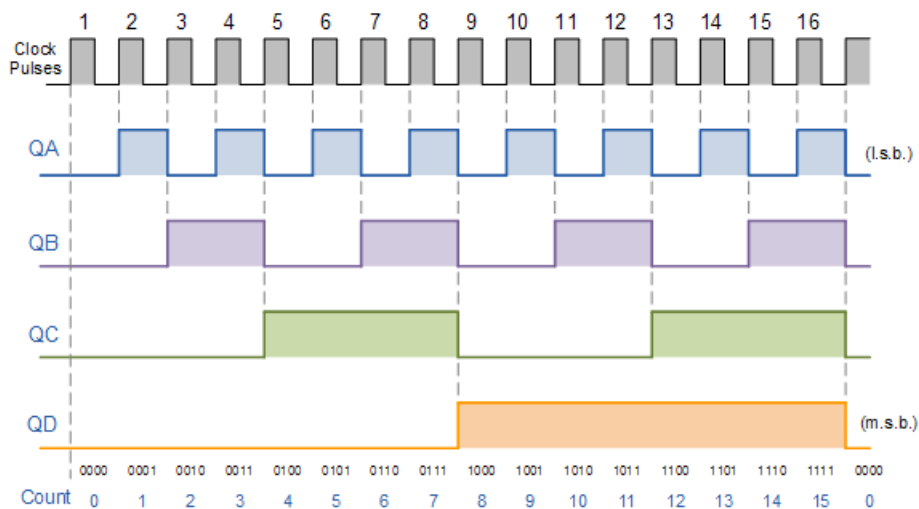
Bij **synchronische tellers** wordt het externe kloksignaal verbonden met ALLE klokingangen van de individuele flipflops die deel uitmaken van de tellerketen. Op deze manier krijgt het fenomeen van 'propagation delay' geen kans. Door dit opzet worden alle flipflops op hetzelfde moment getriggerd zodat er een strakke tijdrelatie bestaat tussen alle signaaltoestanden van de teller. Anders gezegd: de verandering van de signaaltoestanden van de telleruitgangen gebeurt *synchroon* met het extern aangelegde kloksignaal. Er is hier dus geen sprake meer van een zogenaamd 'ripple effect' want alle uitgangbits veranderen exact op hetzelfde tijdstip bij de neergaande of opgaande flank van het gemeenschappelijke kloksignaal.

4.6.2 Synchronische opteller



△ Fig.4.52 | Principeschema van een synchrone opteller.
Bron: <https://www.electronics-tutorials.ws>

In figuur 4.52 zien we duidelijk dat alle klokingangen van de JK-flipflops gevoed worden via een gemeenschappelijke externe klok. Door J en K met elkaar te verbinden bevinden de JK-flipflops (FFB, FFC en FFD) zich in 'toggle mode': bij een logisch '1' niveau zal de uitgang omklappen naar het andere signaalniveau. Alleen bij flipflop FFA zijn J en K beiden verbonden met een *vaste* logische '1' om het omklappen op commando van het kloksignaal mogelijk te maken. Q_A is de minst betekenende bit (LSB) en Q_D is de meest betekenende bit (MSB). Deze configuratie zorgt er voor dat er een *vooropgesteld* toestandenpatroon gegenereerd wordt: voor elke klokpuls staat één bepaalde tellertoestand.



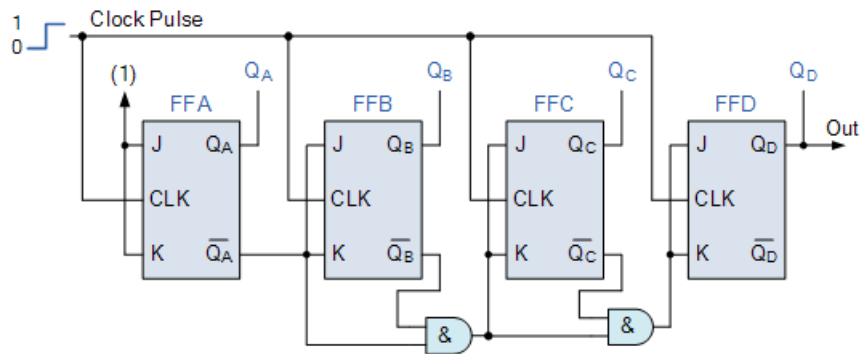
△ Fig.4.53 | Tellertoestanden van een synchrone teller.
Bron: <https://www.electronics-tutorials.ws>

De J en K ingangen van flipflop FFB zijn direct verbonden met de uitgang Q_A van flipflop FFA. De J en K ingangen van flipflop FFC en FFD worden aangestuurd door aparte **AND-poorten**. Deze AND-poorten worden gevoed met ingangsignalen en uitgangsignalen van de vorige trap. De bijkomende poortlogica zorgt voor de juiste JK-signalen voor de volgende trap.

Omdat we elke JK-flipflop in staat stellen om te *complementeren* op basis van het al dan niet 'hoog' zijn van voorliggende flipflop-uitgangen, kunnen we hetzelfde uitgangspatroon genereren zoals bij het asynchrone ontwerp maar dan zonder het nadelige 'ripple effect'. Er is hier ook geen sprake meer van 'propagation delay' zodat we tellerschakelingen kunnen ontwerpen op veel hogere klokfrequenties.

De teller van figuur 4.52 telt bij elke klokpuls op van $(0000)_2$ tot $(1111)_2$ en dit patroon blijft hij herhalen zolang het kloksignaal aanwezig is. Dit tellertype is bekend onder de naam: **'4-bit synchrone opteller'**.

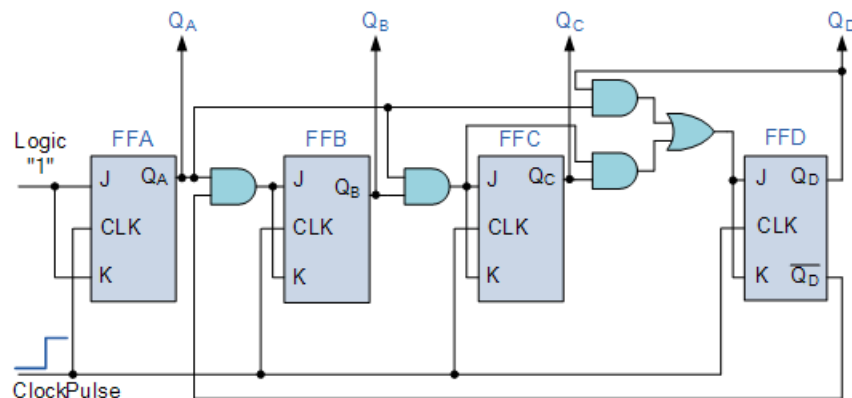
4.6.3 Synchrone afteller



△ Fig.4.54 | Principeschema van een synchrone afteller.
Bron: <https://www.electronics-tutorials.ws>

Het is vrij eenvoudig om de schakeling van de 4-bit synchrone opteller om te vormen naar een **4-bit synchrone afteller**. We realiseren dit door de AND-poorten aan te sluiten op de *inverse* uitgangen ($\bar{Q}_B$ en $\bar{Q}_C$) van de respectievelijke flipflops. De teller zal nu starten bij $(1111)_2$ en telt dan, bij elke klokpuls, af naar $(0000)_2$. Zolang het kloksignaal aanwezig is, zal deze telcyclus zich blijven herhalen.

4.6.4 Synchrone decadeteller



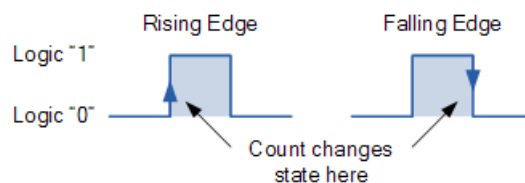
△ Fig.4.55 | Principe van een synchrone decadeteller.
Bron: <https://www.electronics-tutorials.ws>

We maken gebruik van een standaard synchrone opteller met wat bijkomende poortlogica om een **4-bit synchrone decadeteller** te realiseren die telt van $(0)_{10}$ tot $(9)_{10}$.

Flipflop FFA van figuur 4.55 complementeert bij elke klokovertgang. De extra AND-poorten detecteren de tellertoestand $(1001)_2$ en doen flipflop FFD complementeren bij de volgende klokovertgang. De teller wordt hierdoor gereset en het tellen kan weer van voor af aan beginnen.

4.6.5 Triggeren van tellers

Binaire tellerschakelingen maken gebruik van **flankgetriggerde** flipflops waarvan de toestanden veranderen tijdens de **positieve** (stijgende flank) of **negatieve flank** (dalende flank) van het kloksignaal. Deze flanktriggering resulteert steeds in één enkele 1-bit optelling wanneer de status van de klokingang verandert.



△ Fig.4.56 | Positieve en negatieve flank.
Bron: <https://www.electronics-tutorials.ws>

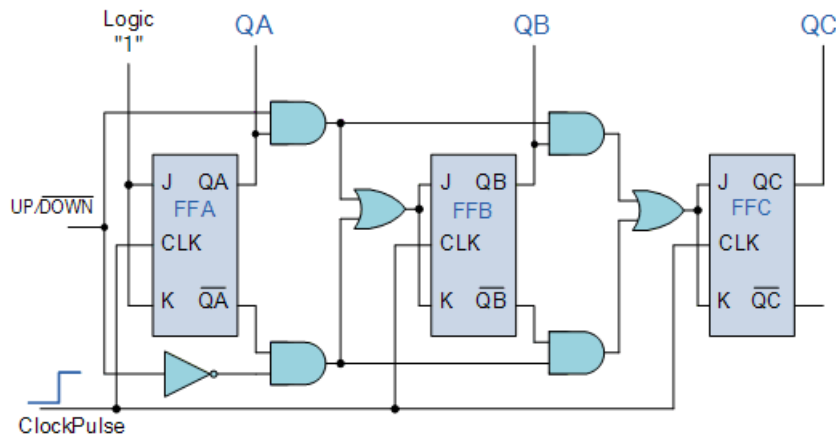
Let op!

Meestal triggeren **synchrone** tellers op de **positieve** flank van het kloksignaal en **asynchrone** (ripple) tellers op de **negatieve** flank van het kloksignaal.



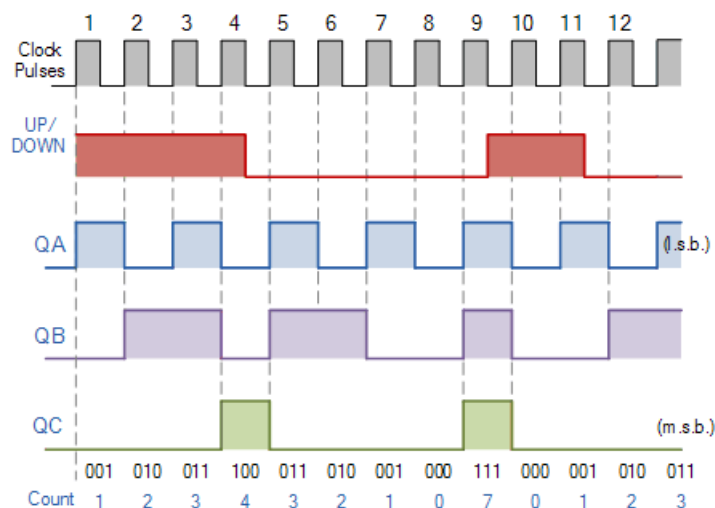
4.6.6 Synchrone bidirectionele tellers

Bidirectionele tellers worden ook wel **up/down counters** genoemd. Ze zijn in staat om zowel OP te tellen als AF te tellen op elk willekeurig moment van de telcyclus. Een bijkomende *controle-ingang* (UP/DOWN) maakt dit alles mogelijk.



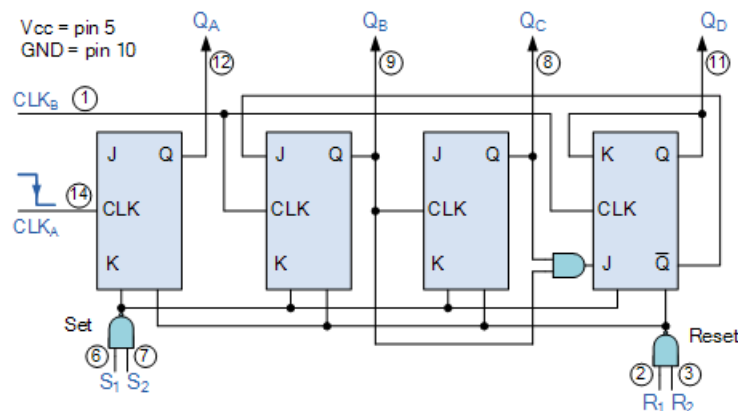
△ Fig.4.57 | Principe van een synchrone up/down counter
Bron: <https://www.electronics-tutorials.ws>

Het schema van figuur 4.57 toont een eenvoudige synchrone 3-bit up/down counter die gebruik maakt van JK-flipflops. De flipflops zijn met behulp van bijkomende poortlogica zo geschakeld dat ze op het juiste moment complementeren. Deze 3-bit teller kan optellen van $(000)_2$ tot $(111)_2$ en terug aftellen van $(111)_2$ tot $(000)_2$. Het tijddiagram van figuur 4.58 illustreert de werking.



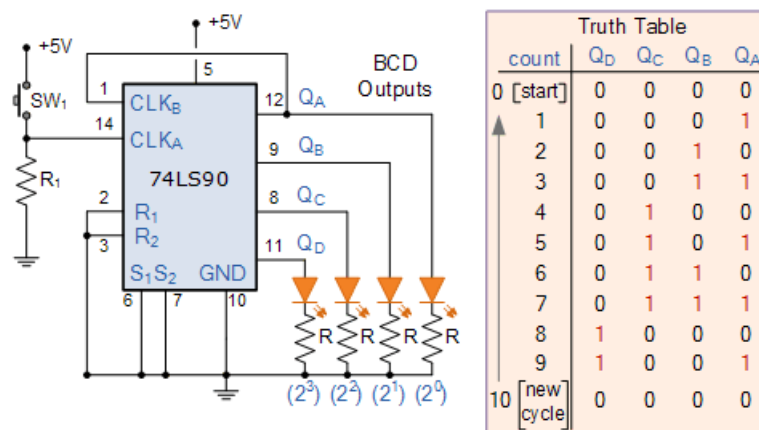
△ Fig.4.58 | Tijddiagram 3-bit up/down counter
Bron: <https://www.electronics-tutorials.ws>

4.7.2 74LS90 BCD-teller



△ Fig.4.60 | Principeschema van een 74LS90 BCD-teller.
Bron: <https://www.electronics-tutorials.ws>

Het interne schema van het **74LS90** IC is in principe een modulo-10 decadeteller die een BCD-code genereert op zijn uitgangen. Het IC herbergt 4 JK-flipflops en bezit 2 kloklijnen (CLK_A en CLK_B). De modulo-10 is het resultaat van de cascade van een 2-deler en een 5-deler. De teller triggert op de *negatieve* flank van het kloksignaal. De bijkomende ingangspinnen R₁ en R₂ *resetten* de teller op (0)₁₀ en de ingangspinnen S₁ en S₂ *setten* de teller op de maximale tellerwaarde (9)₁₀. Figuur 4.61 toont een praktische realisatie die gebruik maakt van dit IC.

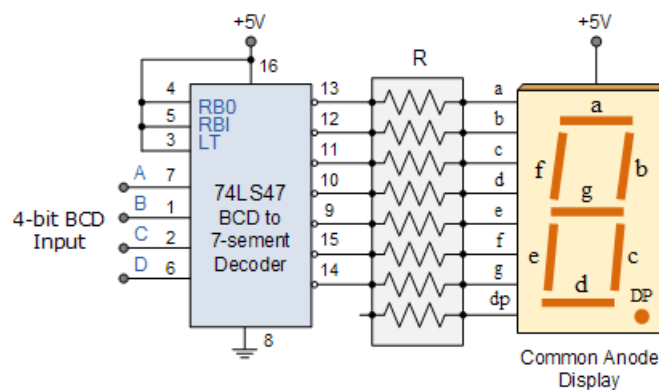


△ Fig.4.61 | Praktische realisatie met 74LS90.
Bron: <https://www.electronics-tutorials.ws>

Telkens je drukknop SW_1 loslaat, nadat je hem ingedrukt hebt, zal de tellerinhoud met 1 toenemen. Wanneer je de drukknop een 10^e keer bediend hebt, worden de uitgangen Q_A t.e.m. Q_D terug op $(0)_{10}$ gezet en kan een nieuwe telcyclus beginnen. We kunnen de toestand van de BCD-teller via LED's zichtbaar maken waarbij een oplichtende LED staat voor een zeker gewicht (1-2-4-8). Wanneer we willen gebruik maken van een *7-segment display* dan zullen we eerst de BCD-code moeten omzetten met een **decoder** (74LS47) die de juiste segmenten dan kan aansturen.

4.7.3 BCD naar 7-segment decoder

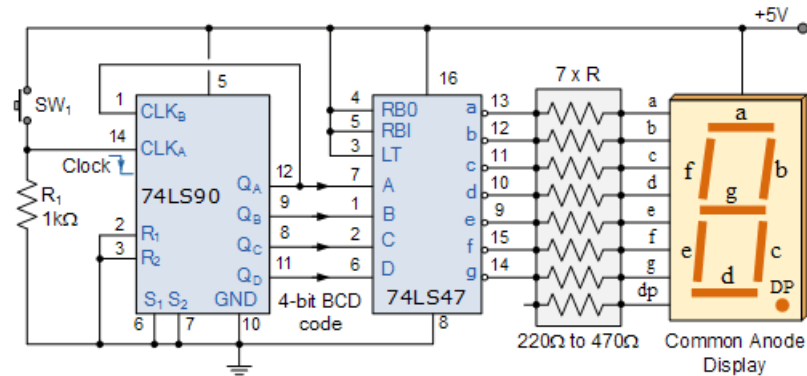
Gelukkig hoeven we de **BCD naar 7-segment decoder** niet zelf te ontwerpen en kunnen we gebruik maken van het populaire IC **74LS47**. Dit decoder-IC bezit 4 ingangen voor de BCD-digits en 7 uitgangen voor elk segment van het display. Merk op dat een standaard 7-segment display 8 aansluitingen heeft: één voor elk LED-segment en één gemeenschappelijke (COM) aansluiting.



△ Fig.4.62 | BCD naar 7-segment 74LS47 decoder
Bron: <https://www.electronics-tutorials.ws>

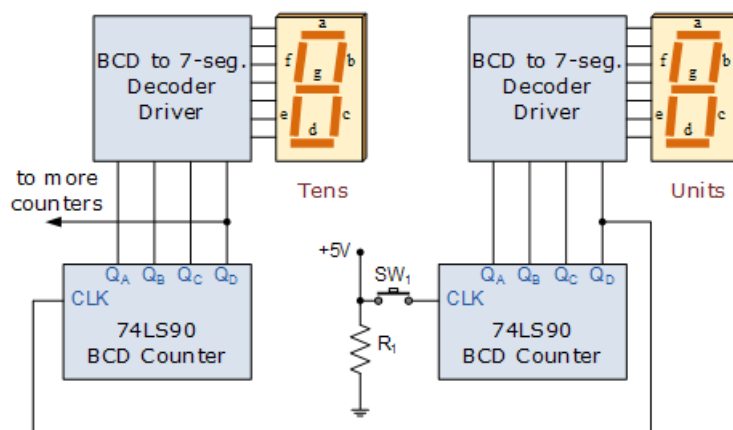
De 74LS47 display decoder zet de BCD-code om naar de noodzakelijke signalen om de juiste LED-segmenten te doen oplichten. Deze decoder bezit *actief lage uitgangen* en is ontworpen om een **common-anode display** aan te sturen zodat we een logisch '0' signaal nodig hebben om een LED-segment te doen oplichten. Bij een logisch '1' signaal zal het LED-segment doven. Voor normale werking plaatsen we pinnen 3, 4 en 5 op een logisch '1' niveau.

De schakeling van figuur 4.63 toont een eenvoudige 0 tot 9 teller die gebruik maakt van de **74LS90** BCD teller en de **74LS47** 7-segment display driver. Door de 4 uitgangen (Q_A , Q_B , Q_C en Q_D) van de BCD-teller aan te sluiten op de respectievelijke ingangspinnen (A, B, C en D) van de decoder kunnen we de telsequentie visualiseren met een 7-segment display. Telkens we de drukknop SW_1 loslaten zal de tellerwaarde op het display met één verhogen.



△ Fig.4.63 | 7-segment display en BCD-decoder.
Bron: <https://www.electronics-tutorials.ws>

Merk op dat in serie met elk LED-segment een voorschakelweerstand in de schakeling is opgenomen. Deze methode om de stroom doorheen de LED's te begrenzen tot zo'n 12 à 20mA geniet de voorkeur. Het gebruik van één enkele gemeenschappelijke voorschakelweerstand is af te raden. Om verder dan $(10)_{10}$ te tellen, dienen we twee BCD-decadetellers in **cascade** te schakelen. Alzo vormen we een 2-digit BCD-teller en kunnen we tellen van $(00)_{10}$ tot $(99)_{10}$. De 0 tot 99 tellerschakeling van figuur 4.64 toont het principe van dit cascadeschakelen.



△ Fig.4.64 | Uitbreiding naar 2 digits.
Bron: <https://www.electronics-tutorials.ws>

4.7.4 74LS90 datasheet



DECADE COUNTER; DIVIDE-BY-TWELVE COUNTER; 4-BIT BINARY COUNTER

The SN54/74LS90, SN54/74LS92 and SN54/74LS93 are high-speed 4-bit ripple type counters partitioned into two sections. Each counter has a divide-by-two section and either a divide-by-five (LS90), divide-by-six (LS92) or divide-by-eight (LS93) section which are triggered by a HIGH-to-LOW transition on the clock inputs. Each section can be used separately or tied together (Q to CP) to form BCD, bi-quinary, modulo-12, or modulo-16 counters. All of the counters have a 2-input gated Master Reset (Clear), and the LS90 also has a 2-input gated Master Set (Preset 9).

- Low Power Consumption . . . Typically 45 mW
- High Count Rates . . . Typically 42 MHz
- Choice of Counting Modes . . . BCD, Bi-Quinary, Divide-by-Twelve, Binary
- Input Clamp Diodes Limit High Speed Termination Effects

FUNCTIONAL DESCRIPTION

The LS90, LS92, and LS93 are 4-bit ripple type Decade, Divide-By-Twelve, and Binary Counters respectively. Each device consists of four master/slave flip-flops which are internally connected to provide a divide-by-two section and a divide-by-five (LS90), divide-by-six (LS92), or divide-by-eight (LS93) section. Each section has a separate clock input which initiates state changes of the counter on the HIGH-to-LOW clock transition. State changes of the Q outputs do not occur simultaneously because of internal ripple delays. Therefore, decoded output signals are subject to decoding spikes and should not be used for clocks or strobes. The Q₀ output of each device is designed and specified to drive the rated fan-out plus the CP₁ input of the device.

A gated AND asynchronous Master Reset (MR₁ • MR₂) is provided on all counters which overrides and clocks and resets (clears) all the flip-flops. A gated AND asynchronous Master Set (MS₁ • MS₂) is provided on the LS90 which overrides the clocks and the MR inputs and sets the outputs to nine (HLLH).

Since the output from the divide-by-two section is not internally connected to the succeeding stages, the devices may be operated in various counting modes.

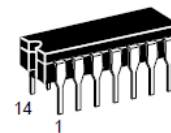
LS90

- BCD Decade (8421) Counter — The CP₁ input must be externally connected to the Q₀ output. The CP₀ input receives the incoming count and a BCD count sequence is produced.
- Symmetrical Bi-quinary Divide-By-Ten Counter — The Q₃ output must be externally connected to the CP₀ input. The input count is then applied to the CP₁ input and a divide-by-ten square wave is obtained at output Q₀.
- Divide-By-Two and Divide-By-Five Counter — No external interconnections are required. The first flip-flop is used as a binary element for the divide-by-two function (CP₀ as the input and Q₀ as the output). The CP₁ input is used to obtain binary divide-by-five operation at the Q₃ output.

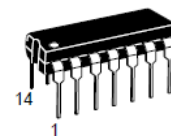
**SN54/74LS90
SN54/74LS92
SN54/74LS93**

**DECADE COUNTER;
DIVIDE-BY-TWELVE COUNTER;
4-BIT BINARY COUNTER**

LOW POWER SCHOTTKY



**J SUFFIX
CERAMIC
CASE 632-08**



**N SUFFIX
PLASTIC
CASE 646-06**



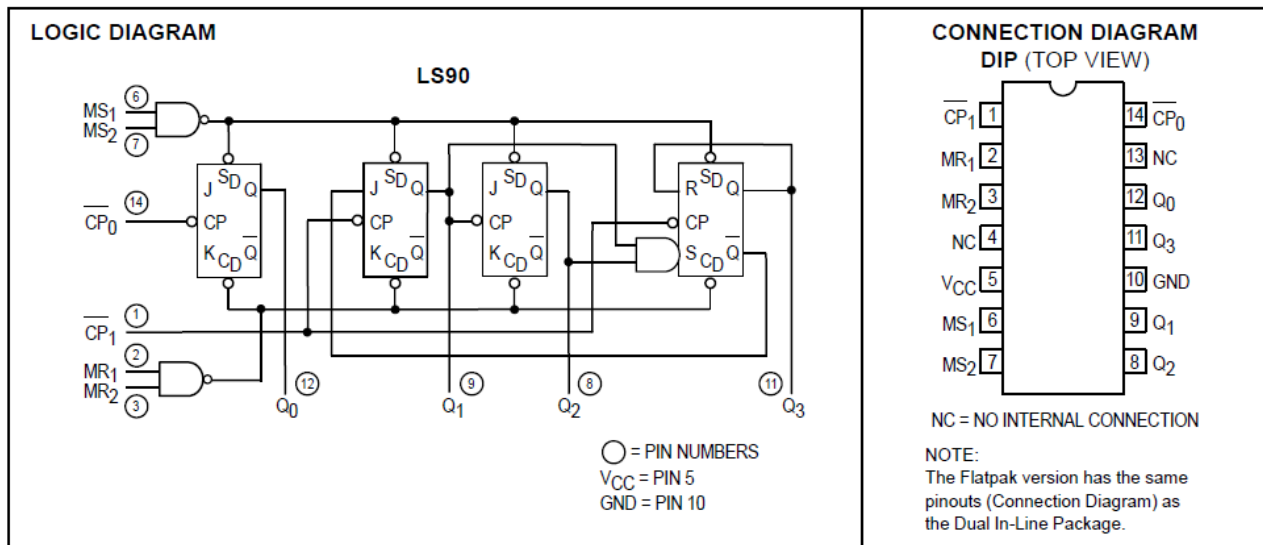
**D SUFFIX
SOIC
CASE 751A-02**

ORDERING INFORMATION

SN54LSXXJ	Ceramic
SN74LSXXN	Plastic
SN74LSXXD	SOIC

PIN NAMES

CP ₀	Clock (Active LOW going edge) Input to +2 Section
CP ₁	Clock (Active LOW going edge) Input to +5 Section (LS90), +6 Section (LS92)
CP ₁	Clock (Active LOW going edge) Input to +8 Section (LS93)
MR ₁ , MR ₂	Master Reset (Clear) Inputs
MS ₁ , MS ₂	Master Set (Preset-9, LS90) Inputs
Q ₀	Output from +2 Section (Notes b & c)
Q ₁ , Q ₂ , Q ₃	Outputs from +5 (LS90), +6 (LS92), +8 (LS93) Sections (Note b)



LS90 MODE SELECTION

RESET/SET INPUTS				OUTPUTS			
MR ₁	MR ₂	MS ₁	MS ₂	Q ₀	Q ₁	Q ₂	Q ₃
H	H	L	X	L	L	L	L
H	H	X	L	L	L	L	L
X	X	H	H	H	L	L	H
L	X	L	X	Count			
X	L	X	L	Count			
L	X	X	L	Count			
X	L	L	X	Count			

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care

LS90 BCD COUNT SEQUENCE

COUNT	OUTPUT			
	Q ₀	Q ₁	Q ₂	Q ₃
0	L	L	L	L
1	H	L	L	L
2	L	H	L	L
3	H	H	L	L
4	L	L	H	L
5	H	L	H	L
6	L	H	H	L
7	H	H	H	L
8	L	L	L	H
9	H	L	L	H

NOTE: Output Q₀ is connected to Input CP₁ for BCD count.

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V _{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T _A	Operating Ambient Temperature Range	54 74	-55 0	25 25	125 70	°C
I _{OH}	Output Current — High	54, 74			-0.4	mA
I _{OL}	Output Current — Low	54 74			4.0 8.0	mA

Laboproef 8

Realiseer een **BCD-decadeteller** die telt van (0)₁₀ t.e.m. (9)₁₀ m.b.v. het IC 74LS90. Maak gebruik van een BCD naar 7-segment decoder IC 74LS47 om een common anode 7-segment display aan te sturen.

Doe eerst een computersimulatie vooraleer je de schakeling in praktijk brengt op een breadboard.

Het induwen van een NO-druknop levert de klokpuls. Zorg er voor dat je door het bedienen van de drukknoop een negatieve flank levert aan de klokingang.