

Praktische Opdracht Schakelingen: Stoplichtenkruising



Inhoud

Inleiding en algemene info.....	2
Aanpak en opzet.....	2
Inleveren en deadline	2
Beoordeling.....	3
Basisopdrachten.....	4
Opdracht 1: Kennismaken met de Counter	4
Opdracht 2: Nog verder tellen? Maak een 16-bits counter	5
Opdracht 3: Zelf de grenzen van de counter bepalen.....	5
Opdracht 4: Een enkelvoudig stoplicht gedreven door een counter	6
Opdracht 5: Een basiskruising met 2 (groepen) stoplichten.....	8
Eindopdracht.....	9
Aanpak	9
Beoordeling.....	10
Suggesties voor uitbreidingen.....	10
<i>Verantwoording en licentie</i>	11

Inleiding en algemene info

Ontwerp met MMLogic een logische schakeling die de stoplichten op een kruising aanstuurt. Je doet deze opdracht alleen of met zijn tweeën.

Aanpak en opzet

Op de volgende bladzijden van dit boekje vind je eerst een aantal inleidende opgaven die je kennis laten maken met de benodigde onderdelen en inzichten. Na het voltooien van opdracht 5 heb je al een eenvoudige kruising met 2 stoplichten gemaakt.

Na het maken van deze basisopgaven ga je zelf een uitgebreidere kruising bedenken en implementeren. Je vindt in dit boekje een aantal tips voor uitbreidingen, je wordt aangemoedigd creatief te zijn en een interessante kruising te ontwerpen.

Inleveren en deadline

Je levert de volgende bestanden in (in een .zip bestand, zie verderop):

- De MMLogic bestanden (.lgi) van alle schakelingen (opdr. 1 t/m 5 + eigen uitbreidingen).
 - Het is handig om alle deelopdrachten van opdracht 1 t/m 5 in 1 MMLogic bestand te maken
 - Maak een apart MMLogic bestand voor je definitieve kruising
 - Een verslag bestaande uit:
 - De antwoorden op basisopgaven 1 t/m 5 (+ screenshots van de bij deze opgaven gemaakte schakelingen)
 - Uitwerking en beschrijving van je uitbreidingen op de opdracht
 - Een toestandsdiagram (waarheidstabel) van je uiteindelijke schakeling
- Maak een .zip van alle PO bestanden inclusief je verslag en logboek en lever het in via de opdracht in de Magister ELO. Per groepje hoeft de opdracht maar 1 keer te worden ingeleverd.

LET OP: dus geen losse bestanden uploaden, maar alles in 1 .zip (Weet je even niet meer hoe je een .zip maakt? Klik met de rechtermuisknop op het mapje met alle in te leveren bestanden erin, wijs "Kopiëren naar" aan en klik vervolgens op Gecomprimeerde map. Upload deze gecomprimeerde map naar de ELO)

Uiterlijke inleverdatum: maandag 24 november

Beoordeling

Als je opdrachten 1 t/m 5 helemaal correct hebt gedaan en je verslag aan de bovenstaande eisen voldoet heb je maximaal een 4. Je kunt na het inleveren gevraagd worden om een toelichting op de gemaakte opgaven, om te laten zien dat je ze zelf hebt gemaakt en begrepen.

Om hoger te halen dan een 4 (en dat wil je natuurlijk) moet je de opdracht zelf uitbreiden. Enkele suggesties voor uitbreidingen vind je hieronder. De keuze voor de uitbreiding is helemaal vrij en mag zo uitgebreid als je wilt. Documenteer je uitbreidingen in je verslag: Beschrijf hoe de uitbreiding werkt en wat het nut/doel ervan is. Bij de beschrijving hoort ook een toestandsdiagram (waarheidstabel) van je uiteindelijke schakeling.

Kijk op de volgende bladzijde voor de basisopdrachten.

Basisopdrachten

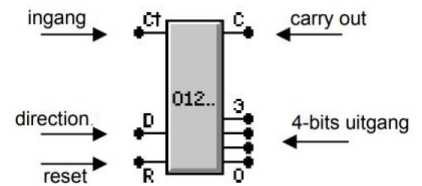
In deze basisopdrachten maak je kennis met de principes en componenten die je nodig hebt om een stoplichtkruising te maken.

Opdracht 1: Kennismaken met de Counter


Het “hart” van elke stoplichtschakeling wordt gevormd door de Counter. Dit is een component dat (de naam zegt het al) kan tellen. De standaardversie is een 4-bits counter (deze kan van 0 tot 15 tellen, dus 2^4 verschillende waarden).

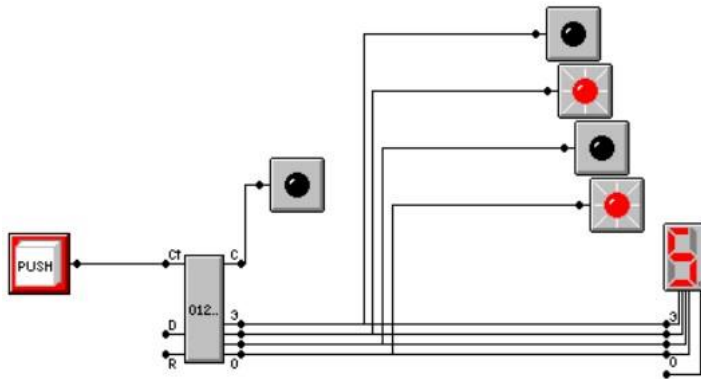
Hier rechts zie je hem staan. In de basis werkt hij als volgt:

- De counter heeft altijd een waarde (0-15 voor de 4-bits versie) en deze waarde staat continu als binair getal op de 4 uitgangen.
- Als je een puls geeft op de ingang linksboven, verhoogt de counter zijn waarde met 1 - (De overige in- en uitgangen komen verderop aan bod)



Opdracht 1a:

Bouw de volgende schakeling na (gebruik deze knop op het palet: ). De push-knop krijg je door een schakelaar toe te voegen en deze te wijzigen in een “momentary” versie via de properties.



Speel even met de counter (door op de push knop te drukken) om een gevoel voor zijn werking te krijgen. Zie je (dankzij de 4 lampjes) het binaire getal van de uitgang van de counter matchen met het hexadecimale getal van het display? Dan zie je hoe de counter werkt.

Vraag 1b:

Wanneer geeft de carry-out (lampje aan de c-uitgang rechtsboven van de counter) een signaal? Waarom zou dat zijn?

Opdracht 1c:

Maak nu een nieuwe schakeling met een nieuwe counter, maar pas deze aan (via de properties) in een 8-bit counter. Je hoeft niet per se ledjes aan te sluiten, je kunt hem uitlezen met behulp van 2 hexadecimale cijfer-displays.

Vraag 1d:

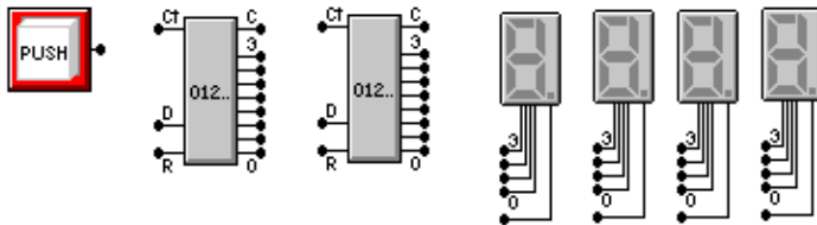
Wat is de limiet van de 8-bits counter? (Dus: wanneer geeft de carry-out een signaal bij de 8-bits counter). Je kunt het testen, maar je kunt het nog beter beredeneren (scheelt je een hoop geklik)

Opdracht 2: Nog verder tellen? Maak een 16-bits counter

In de vorige opdracht ben je begonnen met een 4-bits counter en heb je deze daarna veranderd in een 8-bits counter. Zoals je hebt gezien is er geen optie voor een 16-bit counter. De counter is echter zo ontworpen dat je die makkelijk zelf kunt maken.

Opdracht 2a:

Bedenk zelf hoe je een 16-bit counter kunt maken. Je mag alleen de volgende componenten gebruiken:



Hint: Je ziet dat er maar 1 push-knop is. De truuik is om de 2 counters op een slimme manier aan elkaar te verbinden. Denk goed na hoe!

Opdracht 2b:

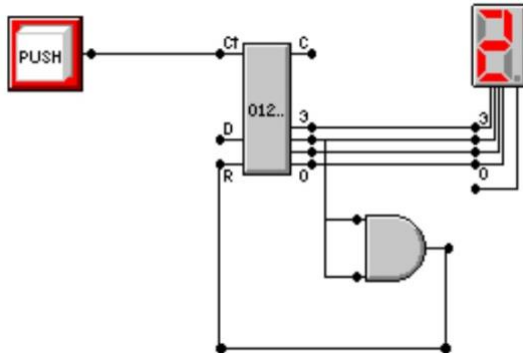
We hebben tot nu toe de D- en R-ingangen van de counter (linksonder) nog niet gebruikt. De D staat voor Direction en de R voor Reset. Voeg een pushknop en een schakelaar toe aan je 16-bits counter van opdracht 2a om te zorgen dat de 16-bits counter gerest kan worden en dat hij 2 kanten op kan tellen.



Opdracht 3: Zelf de grenzen van de counter bepalen

Leuk al die tellers, maar voor sommige toepassingen (het ontwerpen van een stoplichtschakeling om maar eens wat te noemen...) is het handig als je zelf de limiet van de counter kunt bepalen.

De onderstaande schakeling is gemaakt met een 4-bits counter. Hierbij is er een truk uitgehaald (met behulp van de AND-poort) om te zorgen dat hij niet doortelt tot en met 15, maar slechts telt tot 4 en dan weer bij 0 begint. De counter telt dus: 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4, etc.



Vraag 3a:

Je zou ook kunnen zeggen dat de counter zichzelf reset op 5 (en dat gaat zo snel dat de 5 niet te zien is. Hij springt van 4 meteen naar 0). Om de schakeling na te kunnen bouwen moet je eerst bepalen hoe je het getal 5 als 4-bits binair getal schrijft. Dus: Hoe schrijf je 5 binair (in 4 bits?)

Vraag 3b:

De uitgang van de AND poort is verbonden met de R(eset) ingang van de counter. De 2 ingangen van de AND zijn verbonden met 2 verschillende uitgangen van de counter. Welke 2 uitgangen van de counter zijn dat? (Hint: Gebruik je antwoord van vraag 3a om erachter te komen, want aan het plaatje kun je het niet zien).

Opdracht 3c:

Bouw de schakeling na en test of hij inderdaad t/m 4 telt en dan steeds reset.

Opdracht 4: Een enkelvoudig stoplicht gedreven door een counter

Je bent nu bekend met de werking van de counter. Tijd om hem aan het werk te zetten om een stoplicht te maken. We beginnen met een enkelvoudig stoplicht. In de volgende opgave gaan we 2 samenwerkende stoplichten op 1 kruising maken.

De counter gaat voor ons bijhouden in welke “toestand” het stoplicht is. Een enkelvoudig stoplicht heeft 3 toestanden: Het is rood, het is groen of het is oranje. Na oranje springt het stoplicht weer op rood en begint de cyclus weer opnieuw.



Dit gedrag willen we even “vastleggen”, zodat we de counter goed kunnen instellen. Hiervoor gaan we een tabel maken. In de tabel koppelen we een waarde/stand van de counter aan een stoplicht-toestand.

Opdracht 4a:

Maak onderstaande tabel na en vul de ontbrekende binaire kolom in.

Stap (decimaal)	Stap (binair)	Stoplicht
0		Rood
1		Groen
2		Oranje
3		Reset

Met de tabel hebben we nu vastgelegd welk licht gaat branden in welke toestand van de counter. Omdat er maar 3 toestanden zijn, hebben we een reset nodig in stand 3, zodat de counter na 2 weer op 0 springt en dus blijft tellen: **0 1 2 0 1 2 0 1 2 0 ...**

Opdracht 4b:

Bouw een schakeling met een counter en een cijferdisplay. Zorg dat de counter zich reset op 3. (Vergelijkbaar met opdracht 3).

Opdracht 4c:

Nu gaan we het stoplicht toevoegen. Voeg 3 leds aan de schakeling van 4b toe en stel via "properties" de kleuren ervan in op rood, groen en oranje, zodat het een stoplicht wordt. We moeten nu de lichten gaan koppelen aan de counter, zodat in de juiste stap het juiste licht aangaat. (Zoals beschreven in de tabel van 4a).

Het groene (stap 1 uit de tabel) en oranje (stap 2) licht kun je rechtstreeks aan de counter aansluiten. Kijk goed naar de binaire getallen die je in de tabel van opdracht 4a hebt ingevuld bij stap 1 en stap 2 om te weten welke uitgang van de counter je voor elk licht nodig hebt.

Het rode licht is lastiger. Hiervoor hebben we een poort nodig. In dit geval komt de NOR poort goed van pas. Je ziet hier de waarheidstabel van de NOR poort:



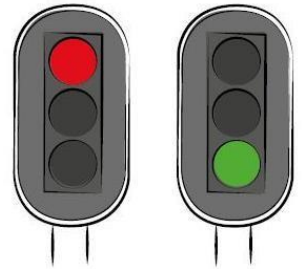
Input 1	Input 2	Output
0	0	1
0	1	0
1	0	0
1	1	0

Het opvallende aan de NOR is, dat er op een hele specifieke plek bij de output een 1 verschijnt. Dit kun je gebruiken om het rode licht aan de counter te hangen. Probeer te beredeneren hoe en sluit dan het rode licht aan met behulp van de NOR. Je maakt de NOR in MMLogic door een OR-poort toe te voegen en deze via de properties in een NOR te veranderen.

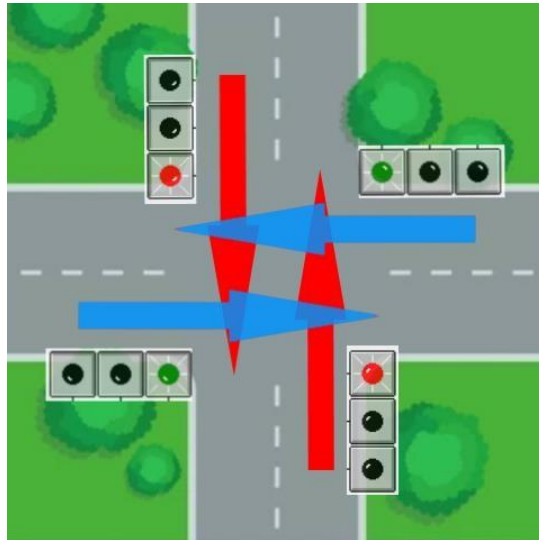
Test je stoplicht. Als het goed is zou hij nu netjes langs de toestanden moeten lopen: ROOD
🚦 GROEN 🚦 ORANJE 🚦 ROOD 🚦 GROEN 🚦 etc, etc.

Opdracht 5: Een basiskruising met 2 (groepen) stoplichten

In opdracht 4 maakte je je een enkelvoudig stoplicht, maar op een kruising zijn er meerdere stoplichten die met elkaar “samenwerken” om het verkeer op de hele kruising in goede banen te leiden. Je kunt al die stoplichten samen als 1 enkele schakeling beschouwen. Dat wil zeggen: we gebruiken voor alle stoplichten op de kruising samen maar 1 counter.



In deze opdracht gaan we een kruising maken met 2 samenwerkende stoplichten:



Je ziet natuurlijk 4 stoplichten staan, maar de 2 stoplichten voor de rode rijrichtingen zijn gelijk aan elkaar en de 2 stoplichten voor de blauwe rijrichtingen zijn ook gelijk aan elkaar. Logisch gezien zijn het dus maar 2 stoplichten, die we dubbel neerhangen.

Zoals je ziet kun je op deze kruising alleen maar rechtdoor. Je wilt vermoedelijk ook mogelijk maken dat je linksaf en rechtsaf kunt slaan. Dat is straks je de meest logische eerste uitbreiding van je kruising. Als je deze opdracht goed begrijpt, dan is het maken van een complexere kruising niet heel lastig.

Opdracht 5a:

Om de cyclus van de stoplichten goed vast te leggen en te koppelen aan de stappen van de counter, gaan we weer een tabel maken. Je ziet dat we voor het extra stoplicht, maar 1 extra

stap nodig hebben. Omdat we nu tot 4 tellen, hebben we 3 bits binaire getallen nodig i.p.v. 2bits zoals in opgave 4. Vul zelf de binaire kolom weer goed in:

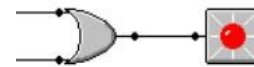
Stap (decimaal)	Stap (binair)			Stoplicht A	Stoplicht B
0				Groen	Rood
1				Oranje	Rood
2				Rood	Groen
3				Rood	Oranje
4				Reset	

Opdracht 5b:

Om nu de stoplichten in de juiste stappen aan te laten gaan, kunnen we een slimme truuk gebruiken. Het mooie aan deze truuk is dat hij goed uitbreidbaar is. Dat wil zeggen: als je straks een uitgebreidere kruising ontwerpt met meer stoplichten en meer stappen, kun je dezelfde truuk gebruiken.

Je mag zelf even nadenken en puzzelen op het aansluiten van de stoplichten. Kom je er niet uit? [Kijk de uitleg van Wt op Youtube](#) om je te helpen.

NB: In het filmpje ben ik een klein stapje vergeten. Je wilt de rode lichten 2x aansluiten (de rode lichten moeten 2 stappen lang aanblijven, zoals je in de tabel kunt zien). Je kunt een input maar 1x aansluiten in MMLogic. Dit is makkelijk op te lossen door een OR poort ervoor te hangen.



Eindopdracht

Aanpak

Je hebt nu alles wat je nodig hebt om zelf een kruising te ontwerpen. Voeg extra rijbanen toe om een complexere kruising te krijgen, of voetgangers of...

Hieronder vind je suggesties voor uitbreidingen en je mag natuurlijk zelf ook dingen verzinnen.

Maak eerst een schets op papier van hoe de kruising eruit komt te zien. Gebruik deze schets om te bepalen welke rijbanen in welke volgorde aan de beurt kunnen komen. (Sommige kunnen vaak tegelijk, dus kies slim!).

Maak vervolgens een waarheidstabel van je kruising en gebruik je kennis van opdracht 5 om deze om te zetten naar een werkende logische schakeling.

Beoordeling

Je wordt aangemoedigd om zo veel mogelijk uitbreidingen te maken natuurlijk.

Ter indicatie: voor een voldoende wordt verwacht dat je je kruising uitbreidt tot minstens 4 verkeersstromen (t.o.v. de 2 verkeersstromen van opdracht 5. Denk aan linksaf en rechtsaf slaan of voetgangers) en voorziet van een goede waarheidstabel en een achtergrondplaatje ter verduidelijking.

Hogere cijfers zijn er uiteraard voor extra toevoegingen daar bovenop.

Suggesties voor uitbreidingen

Hier volgen enkele suggesties voor uitbreidingen van je schakeling. Er zijn er natuurlijk nog veel meer te verzinnen. Wees creatief!

- Voeg een achtergrond afbeelding toe om de situatie te verduidelijken
- Je verwerkt timing in je schakeling. Een stoplicht staat bijvoorbeeld langer op groen dan op oranje en waarschijnlijk nog langer op rood.
- Maak meerdere rijbanen in je kruising (linksaf, rechtdoor etc.) Verzin zelf een goede rijdvolgorde zodat iedereen aan de beurt komt, maar niet te lang moet wachten. En dat er geen botsingen kunnen plaatsvinden natuurlijk.
- Voeg voetgangerslichten toe, waarbij eerst op een knopje moet worden gedrukt om aan te geven dat er iemand staat voordat het licht eventueel op groen gaat. Als er geen voetgangers staan te wachten (het knopje wordt dus niet ingedrukt), worden de voetgangerslichten overgeslagen in de cyclus.
- Je kunt de stoplichten uitzetten (bijvoorbeeld voor 's nachts of tijdens werkzaamheden) en dan gaat er een oranje licht knipperen bij alle stoplichten.
- Maak een "tikker" die blinden helpt met oversteken. Zo'n tikker heeft een ander geluid bij groen en bij rood.
- Er zijn nog veel meer uitbreidingen te bedenken natuurlijk...

Verantwoording en licentie

Deze opdracht is een aanpassing/uitbreiding van de [projectopdracht Logische Schakelingen van Stercollectie Infomatica](#) (voorheen Enigma). Deze aanpassing is gedaan in lijn met de Creative Commons licentie. Hiermee is dit boekje ook beschikbaar onder diezelfde Creative Commons licentie om te gebruiken en aan te passen. Wt / SGDB / 2022

