

Module SPI Controller

Martin Geertjes (4324285), Jeroen van Uffelen (4232690)

14 januari 2015

Samenvatting
bladiebla

Hoofdstuk 1

inleiding

blaballa..

Hoofdstuk 2

LCD controller

2.1 LCD Controller

Op de LCD zal de huidige tijd, ingestelde wekkertijd, datum en ingeschakelde functies te zien zijn. Een LCD is daar handig voor omdat het veel ontwerp vrijheid biedt. Dat neemt ook mee dat het erg gecompliceerd kan worden. Het LCD dat zal worden gebruikt is van de fabrikant MIDAS, typenummer MC128064B6W-BNMLW. Het betreft een graphical LCD van 128 x 64 pixels met een register die geschreven kan worden. De bibliotheek met characters en de controller om het LCD te schrijven zal extern van deze chip plaatsvinden door middel van een atmega32-16pu. Deze keuze is gemaakt omdat voor de characters niet genoeg ruimte is op de chip. De LCD controller op de chip zal dus alleen de inkomende data moeten omzetten naar een positie waarnaar het geschreven moet worden en een bijbehorend character.

2.2 Specificaties

Naam	Type	Functie
clk	in std_logic	Klok
reset	in std_logic	Reset
ready	in std_logic	
uren	in std_logic_vector(5 downto 0)	data signaal met actuele uren afkomstig van DCF
minuten	in std_logic_vector(6 downto 0)	data signaal met actuele minuten afkomstig van DCF
dagvdweek	in std_logic_vector(2 downto 0)	data signaal met de actuele dag afkomstig van DCF
dagvdmaand	in std_logic_vector(5 downto 0)	data signaal met de actuele dag van de maand afkomstig van DCF
maand	in std_logic_vector(4 downto 0)	data signaal met de actuele maand afkomstig van DCF
jaar	in std_logic_vector(7 downto 0)	data signaal met het actuele jaar afkomstig van DCF
dcf_debug	in std_logic	signaal afkomstig van het dcf component en weergeeft of het DCF signaal ontvangen wordt of niet
menu	in std_logic_vector(2 downto 0)	data signaal die de actuele menu state weergeeft
alarm	in std_logic	buffer signaal dat weergeeft of alarmfunctie in of uitgeschakeld is
geluid_signaal	in std_logic	buffer signaal dat weergeeft of geluidsfunctie in of uitgeschakeld is
licht_signaal	in std_logic	buffer signaal dat weergeeft of lichtfunctie in of uitgeschakeld is
wektijd_uren	in std_logic_vector(5 downto 0)	data signaal met ingestelde wektijd uren
wektijd_min	in std_logic_vector(6 downto 0)	data signaal met ingestelde wektijd minuten
data_out	out std_logic_vector(6 downto 0)	data signaal dat de x,y,c informatie doorgeeft aan de microcontroller
clk_out	out std_logic	clock om microcontroller clock mee te synchroniseren

2.2.1 Gedrag

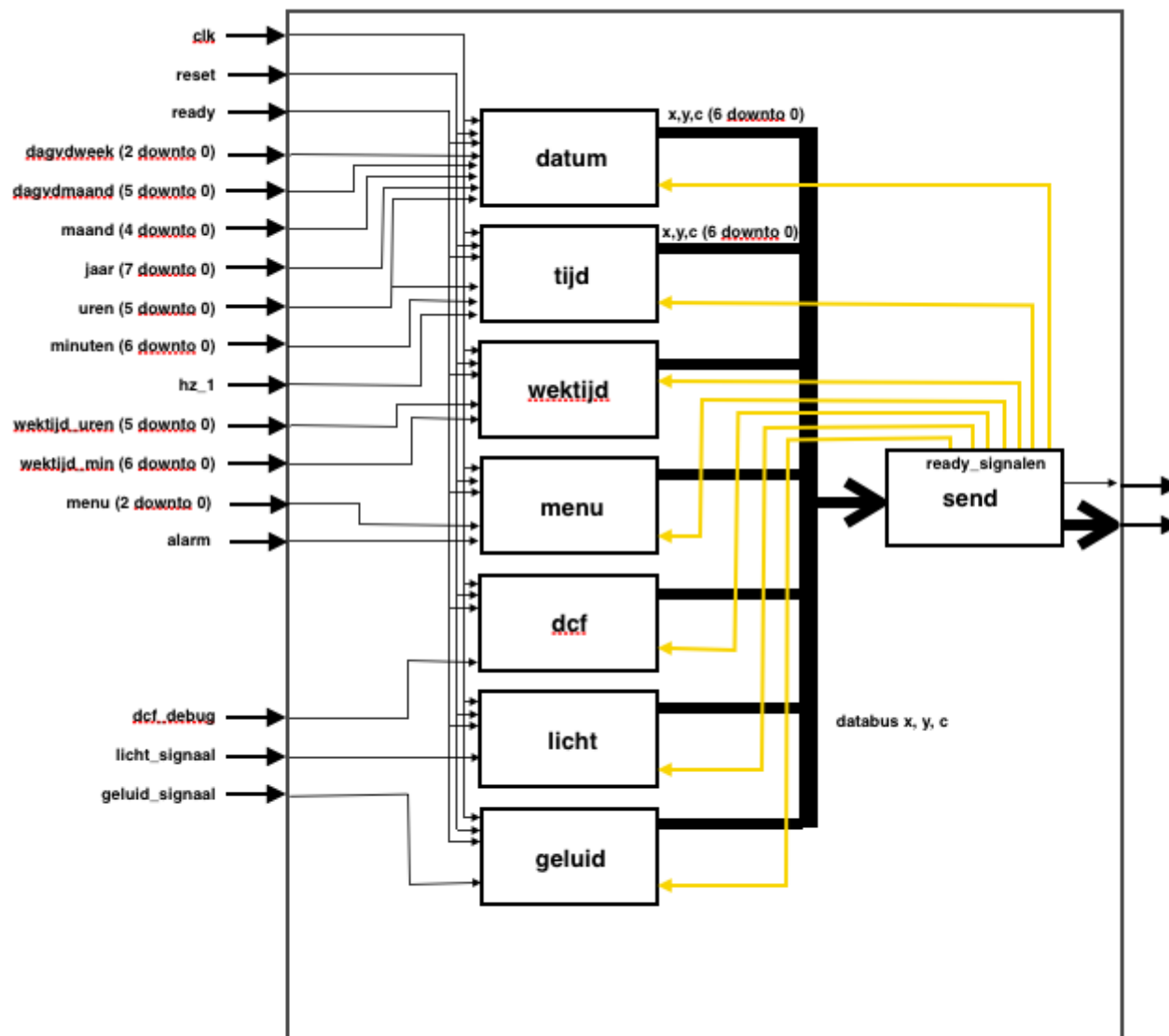
De LCD controller zal na de reset alle informatie die hij binnen krijgt omzetten naar een karakter met bij behorende x en y positie en wegschrijven naar de microprocessor van de LCD. Daarna zal de controller alleen de data die veranderd op de ingangen omzetten en wegschrijven naar de microprocessor om tijd en onnodige acties te besparen.

Het verzenden van de x,y en c gaat door een data signaal van 7 bits samen met een clock_out. Een neergaande klokflank geeft aan dat de data klaar staat om te verzenden zodat er op de opgaande klokflank kan worden gesampled. Zo zal eerst de x, daarna de y en als laatste de c worden verzonden. Het versturen van een karakter duurt dus 3 klokslagen van de clock_out. een klokslag van de clock_out is gelijk aan 2 klokslagen van de ingaande clk.

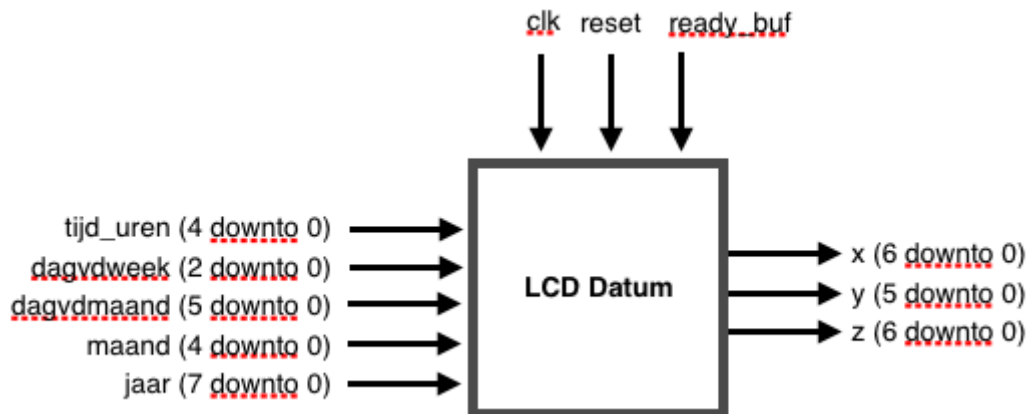
2.3 Functionaliteit

De systemen links (datum, tijd, etc) zorgen per stuk voor het ontvangen van de inkomende informatie en het omzetten naar een x,y positie met een karakter. De x,y en de c zal op de uitgang van het component worden gezet. Het component send_buffer is een MUX en zorgt voor het uitlezen van de x,y en c en zal door middel van de ready signalen aangeven welk signaal hij heeft uitgelezen en naar de zender heeft verstuurd. Zodra de ready laag wordt, weet het desbetreffende component dat de data is uitgelezen en zal daarna nieuwe data klaar zetten. Nadat de mux de data naar de zender heeft gebufferd, zal de zender de signalen een voor een door verzenden naar de microcontroller. Tegelijkertijd zal de zender een clock_out geven, zodra de clock laag wordt staat de data klaar, zodat op de opgaande klokflank de data vanaf de chip kan worden uitgelezen.

2.4 Subsystemen LCD



Figuur 2.1: Toplevel Entity



Figuur 2.2: Entity datum

2.4.1 Datum

Gedrag

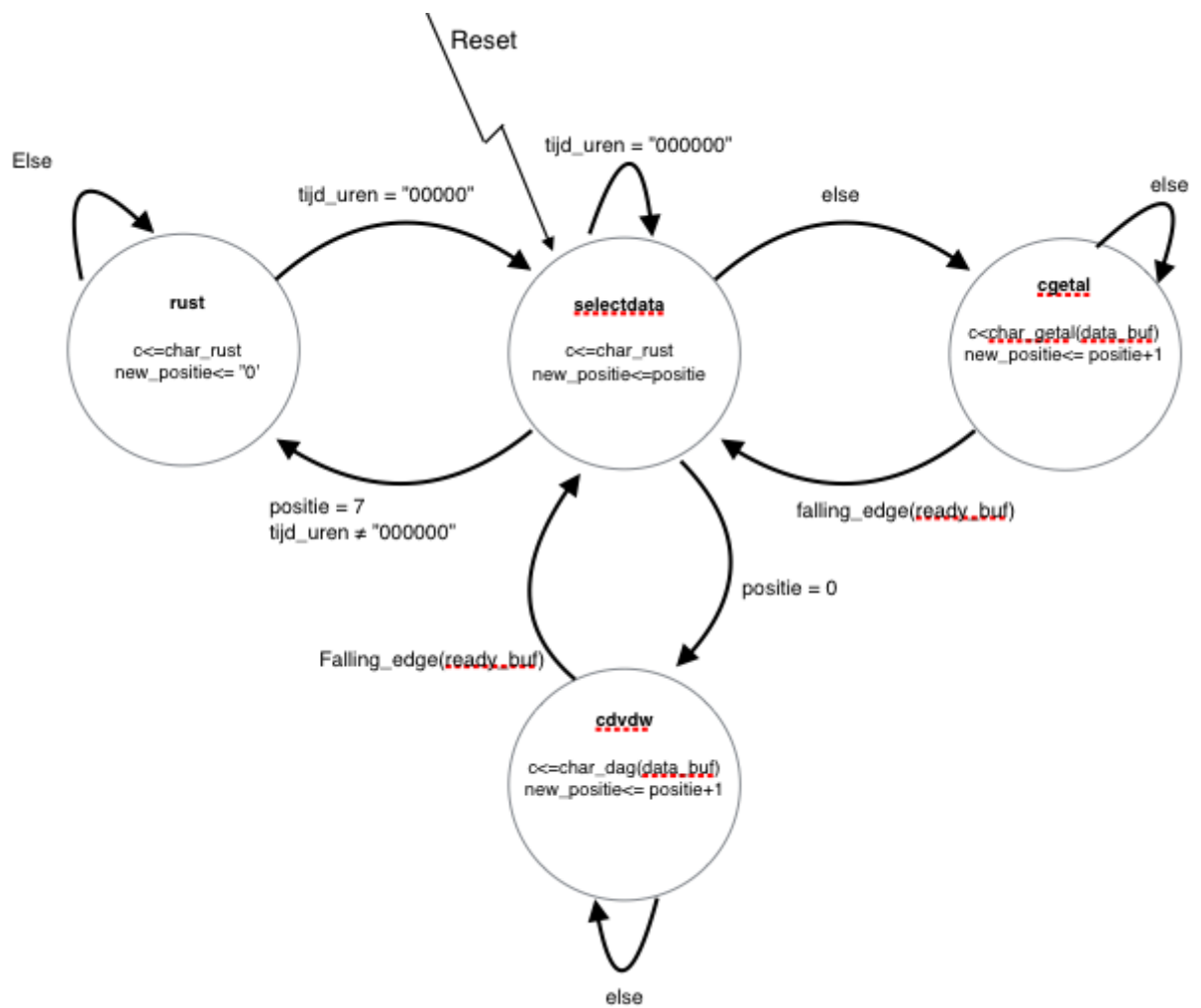
Na een reset of één keer per dag om 12 uur middernacht zal het datum component de data voor de nieuwe datum verzenden naar de zend buffer. De inkomende data is in het binary coded decimal (bcd) formaat. Eerst zal hij de dag van de week doorgeven, daarna de dag van de maand, de maand en daaropvolgend het jaartal. Dat allemaal sequentieel, getriggert op de neergaande klokflank van de ready_buf.

Functionaliteit

Het component werkt met een finite state machine (FSM) waarin de characters worden klaargezet en een counter genaamd positie wordt aangestuurd met daarnaast een apart process om de juiste input te bepalen afhankelijk van de counter.

Na de reset zal de FSM in de selectdata state starten en positie op 0 worden gezet. In het aparte process zal de data_buf worden gekoppeld aan de input die hoort bij positie=0 en zal de x en y bepaald worden. In selectdata zal afhankelijk van de counter (in dit geval 0) worden bepaald of de dag van de week of de getallen van de datum moet worden geschreven. Bij positie=0 zal het karakter van de dag van de week (in state cdvdw) worden klaargezet. Het karakter wordt bepaald door de waarde die op de data_buf staat. Op de neergaande klokflank van het ready_buf signaal zal de FSM terug gaan naar de state selectdata en zal er bij positie 1 worden opgeteld. Daardoor zal hierna het eerste getal van de datum worden geschreven. Dit zal zich herhalen tot positie =7, daarna zal hij naar de rust stand gaan en is de datum geschreven.

Als het middernacht is, dus tijd_uren = '00000', zal er ook een nieuwe datum worden klaargezet. Zodra de positie 7 is zal de machine in state selectdata blijven hangen totdat de tijd_uren ongelijk is aan "00000". Dit om te voorkomen dat het component een uur lang onnodig data blijft verzenden.



Figuur 2.3: FSM Datum

FSM

VHDL code

?? ??

Simulaties

Testen

Tijdens het testen is de

Resultaten

De schakeling werkt correct.

Discussie

2.4.2 Tijd en Wektijd

Gedrag

Na een reset of bij het veranderen van de minuten zal de entity tijd data gaan verzenden. De data die verzonden wordt bevat informatie over welk character geprint moet worden en zijn positie. De characters die moeten worden verzonden worden bepaalt aan de hand van de minuten en uren die de entity in gaan. Dit zal altijd op de volgende volgorde gaan: tientallen uren, eentallen uren, tientallen minuten en als laatste eentallen minuten. Ook zal de module bij een opgaande flank van het één Hz signaal de dubbele punt tussen de uren en minuten aan of uit zetten.

Het component wektijd verzend dezelfde informatie naar het LCD, alzij het op een andere positie. De wektijd zal beginnen met verzenden op het moment dat de wektijd wordt aangepast in het menu. De volgorde van de

Functionaliteit

Het component werkt volgens het Moore principe. Om de seconde wordt een nieuw character naar het LCD gestuurd. Dit character is voor de dubbele punt tussen de uren en minuten. Als er een minuut om is dan zal de nieuwe tijd naar het LCD worden verzonden.

FSM

VHDL code

Simulaties

Testen

Resultaten

Discussie

2.4.3 Wektijd

Gedrag

Functionaliteit

FSM

VHDL code

Simulaties

Testen

Resultaten

Discussie

2.4.4 Menu

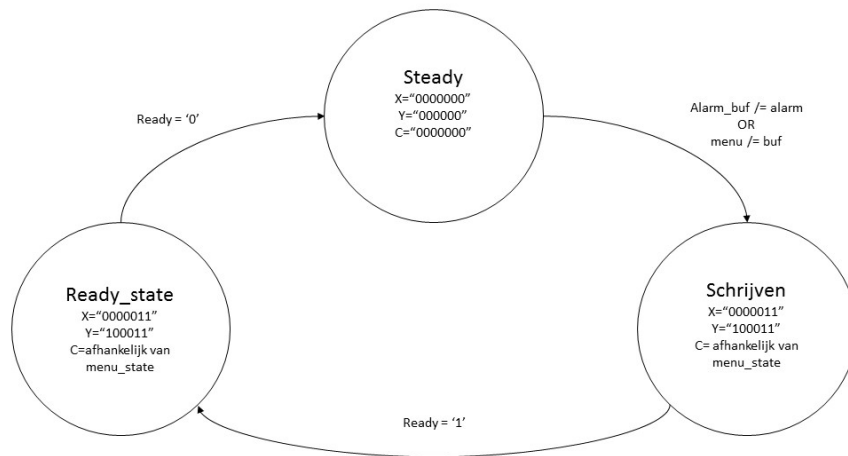
Gedrag

Om de gebruiker feedback te geven in welk gedeelte van het menu de wekker zit wordt dat laten zien op het LCD. Het eerste deel wat het subblok *menu* doet, is aangeven wat er op het LCD geschreven moet worden wat er aangepast moet worden. Het tweede deel is een streep zetten onder het deel wat daadwerkelijk aangepast wordt.

Functionaliteit

In figuur 2.4 is de FSM te zien van het subblok menu. Zolang het menu niet verandert, blijft blijven de uitgangssignalen hetzelfde. Zodra het menu verandert, worden er andere karakters geschreven, welk karakter er geschreven wordt, hangt af van in welk deel van het menu de wekker zit. Zodra het ready-sigitaal hoog wordt, kan de informatie naar de zender gestuurd worden en als het ready-sigitaal dan weer laag wordt, is de informatie verstuurd en wordt de nieuwe state weer de steady-state.

FSM



Figuur 2.4: FSM van het subblok menu

VHDL code

Simulaties

In appendix ?? is de simulatie van het subblok menu te zien.

Testen

Resultaten

Discussie

2.4.5 DCF

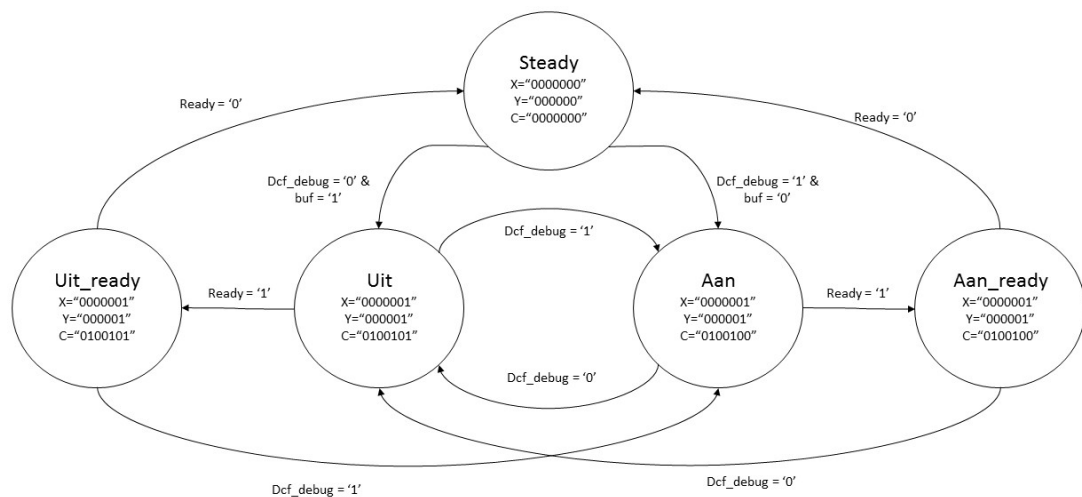
Gedrag

Om te weten of het DCF-signaal ontvangen wordt, wordt er een karakter geschreven op het LCD. Het DCF_debug signaal wat van de DCF-controller komt, geeft aan of het DCF-signaal ontvangen wordt. Om te zorgen dat er niet steeds geschreven hoeft te worden of het signaal ontvangen wordt of niet, is er een buffer geïmplementeerd die het signaal een klokslag onthoudt. Dus alleen als het signaal verandert, hoeft het subblok van de DCF te gaan schrijven.

Functionaliteit

Om duidelijk te maken hoe het subblok DCF_lcd werkt, staat in figuur 2.5 de FSM van het subblok.

FSM



Figuur 2.5: FSM van het subblok dcf_lcd

VHDL code

Simulaties

De resultaten van de simulatie aan het subblok DCF_lcd is te vinden in appendix ??

Testen

Resultaten

Discussie

2.4.6 Geluid en licht

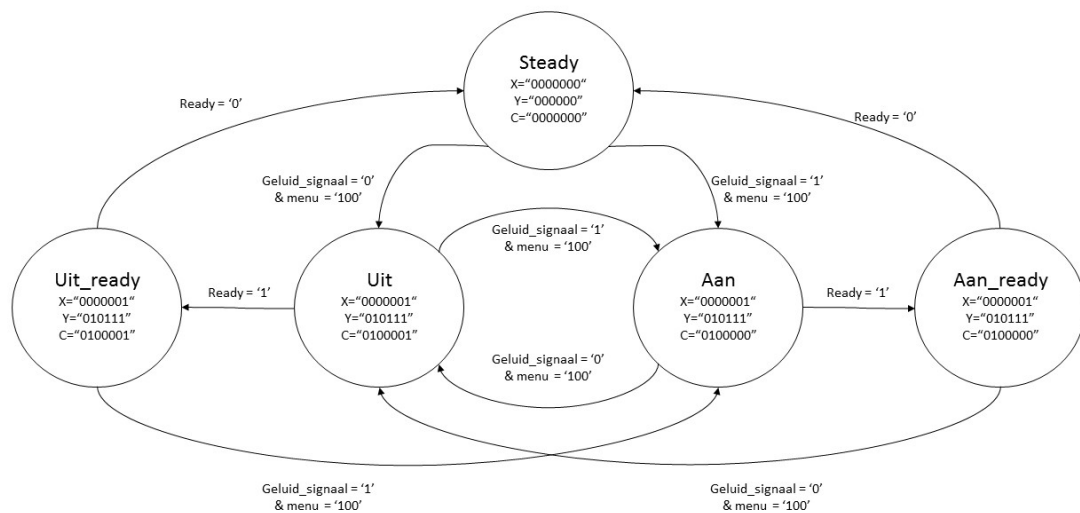
Gedrag

Op het LCD is te zien of het geluid en/of het licht aangaat als het alarm afgaat. Dit is optioneel en kan geregeld worden in het menu. Het signaal menu_state geeft aan in welk deel van het menu de gebruiker zich bevindt.

Functionaliteit

De subblokken licht en geluid lijken veel op elkaar. In figuren 2.6 en 2.7 zijn de FSM van de twee blokken te zien.

FSM



Figuur 2.6: FSM van het subblok geluid

VHDL code

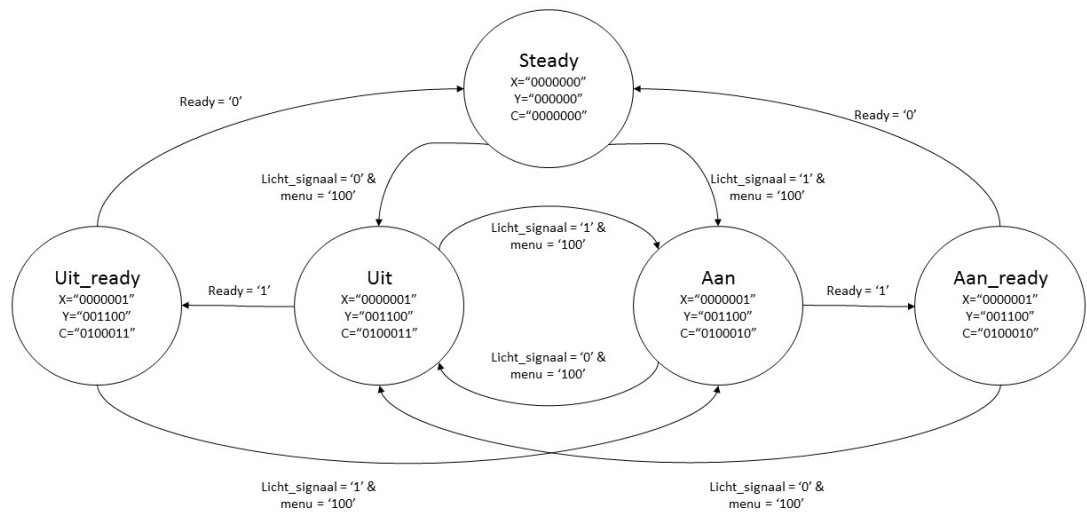
Simulaties

In appendix ?? zijn de resultaten te zien van de subblokken geluid en licht.

Testen

Resultaten

Discussie



Figuur 2.7: FSM van het subblok licht

2.4.7 VHDL code

2.5 Simulatie

2.6 Testen

2.7 Resultaten

2.7.1 Conclusie en discussie

2.8 Bibliografie

Bibliografie

- [1] *Stephen Brown, Zvonko Vranesic* , Fundamentals of Digital Logic with VHDL design, *McGraw-Hill*, Jan 1, 2009
- [2] *Jan M. Rabeay, Anantha Chandrakasan, Borivoje Nikolic* , Digital Integrated Circuits, second edition, *Prentice Hall*, 2003
- [3] *Wikipedia* , Serial Peripheral Interface *Geraadpleegd op 10 november 2014*, http://nl.wikipedia.org/wiki/Serial_Peripheral_Interface