

Mid-term report

Projectgroep A2

17 december 2014

1

SAMENVATTING

INHOUDSOPGAVE

1	Samenvatting	1
2	Inleiding	1
3	Probleemstelling	2
3.1	Projectopgave: 'Ontwerp een chip'	2
3.2	Beschikbare infrastructuur	2
3.3	Functionele eisen	2
3.4	Randvoorwaarden	2
3.5	Plan van aanpak	2
4	Besturing	3
4.0.1	Ultrasone	3
4.1	Ultrasone	3
4.2	Buttons	3
5	Blackbox	4
5.1	Functionele beschrijving	4
5.2	Inputs en outputs	4
5.3	Implementatie	5
6	ALU/PC	6
7	GPU	7
8	VGA	8
9	SPI	9
9.1	Functionele beschrijving	9
9.2	Inputs en outputs	9
9.3	Implementatie	10
10	SD-kaart	11
10.1	SD communicatie	11
10.2	SD commando's & responses	11
10.3	Functionele omschrijving	11
10.4	Inputs en outputs	12
11	Game: Pong	13
12	Discussie	14
12.1	Appendix	15

2

INLEIDING

EPO-3 is het derde project uit de propedeuse van de bachelor opleiding Electrical Engineering aan de TU delft. Het doel van dit project is het ontwerpen van een chip, deze chip kan bijvoorbeeld een spelletje bevatten of als besturingsdoel dienen. Als project groep A5 is er gekozen voor het bouwen van een spelcomputer, hierop kunnen meerdere games gespeeld worden. De verschillende games moeten van een SD-kaart worden geladen voordat ze gespeeld kunnen worden. Het eerste spel dat ontworpen wordt is Pong, mocht er aan het eind van dit project nog tijd overblijven zullen er meerdere spellen volgen. De besturing loopt via ultrasone sensoren die de locatie van iemands hand bepalen. In dit tussentijdse ontwerp rapport zullen al deze onderdelen aanbod komen, daarbij moet vermeld worden dat ze nog niet geheel voltooid en getest zijn.

3

PROBLEEMSTELLING

3.1. PROJECTOPGAVE: 'ONTWERP EEN CHIP'

De opdracht meegegeven door de TU is het ontwerpen van een chip. Hiermee is de gehele opdracht eigenlijk al beschreven, de studenten mogen verder in alle vrijheid de opdracht verder bepalen. Er wordt natuurlijk wel eerst grondig gecontroleerd door de tutores of de opdrachten aan de eisen voldoen en haalbaar zijn. Naast het ontwerpen van de chip mogen er extern onderdelen worden toegevoegd, bijvoorbeeld een beeldscherm of onderdelen voor de besturing.

Als project groep A5 is er gekozen om van de chip een spelcomputer te maken, hierop moeten meerdere spellen gespeeld kunnen worden. Als eerst moet het spelletje 'Pong' te spelen zijn op deze spelcomputer. Aan het einde van het project kunnen eventueel andere spellen ontworpen worden. Extern moet er gezorgd worden voor besturing, weergave en opslag van de spellen. Dit laatste wordt door middel van een SD-kaart gerealiseerd. De besturing vindt plaats via ultrasone sensoren of mocht dit niet naar behoren werken door middel van drukknoppen.

3.2. BESCHIKBARE INFRASTRUCTUUR

In het 4e semester wordt de chip geproduceerd en getest, deze chip wordt door twee projectgroepen gebruikt. Elke groep heeft dus een halve chip voor zijn ontwerp, dit zijn ongeveer 200-250 flipflops. Daarnaast kan iedere groep wel gebruik maken van alle aansluitpunten op de chip, dit zijn er 32. Voor het testen tussen door is er hardware beschikbaar, in de vorm van een Altera FPGA bord. Op dit Altera bord is veel randapparatuur aanwezig, daarnaast kunnen er nog andere externe onderdelen aan dit bord worden gekoppeld.

3.3. FUNCTIONELE EISEN

Als functionele eisen kan er kortom gezegd worden dat er op de chip meerdere spellen gespeeld kunnen worden. Deze spellen moeten bestuurd kunnen worden door middel van ultrasone sensoren. Om te beginnen wordt 'Pong' ontworpen, andere spellen volgen als de projecttijd dit toelaat.

3.4. RANDVOORWAARDEN

Aan het eindresultaat zitten een aantal randvoorwaarden verbonden. De externe onderdelen mogen maximaal 30 á 40 euro kosten, mochten deze niet voorhanden zijn. Er moet rekening worden gehouden met de grootte van de chip. Tot slot dient aan alle tijdschema's te worden voldaan.

3.5. PLAN VAN AANPAK

Het plan van aanpak is een beschrijving van de aanpak van dit project, hierin staan onder andere de doelstellingen en wegen daar naar toe beschreven. Het plan van aanpak is te vinden in appendix A.

4

BESTURING

Ons systeem wordt gespeeld doormiddel van 2 soorten besturing. De Ultrasonen en de Button. De speler kan de modus selecteren doormiddel van een switch. Deze data van de buttons en de ultrasonen worden verwerkt door een Arduino. Er is gekozen voor deze optie omdat de Arduino via SPI werkt. Hierdoor kan de SPI code getest worden en kunnen we het aantal pinnen dat gebruikt wordt laag houden.

4.0.1. ULTRASONEN

Een unieke uitdaging van ons project is de Ultrasonen besturing, de besturing werkt door middel van een (ultrasonen)sensor aan de rechterzijde van de speler. Deze sensor meet met een speel ruimte van 75cm elke 11 milliseconde. Het aansturen van de ultrasonen sensoren gaat doormiddel van een 2ms lange pulse op de IN-pin. Dit is de trigger van de SRF-04(de door ons gekozen ultrasonen sensor). Hierna verzendt de sensor zijn pulse, op dat moment wordt de OUT-pin ook hoog. Deze blijft hoog tot de gereflecteerde pulse weer binnen komt. Deze tijd wordt gedeeld door 2 omdat het geluid zowel de afstand heen als terug moet afleggen. Deze tijd wordt vervolgens geschaald naar afstand door hem door 29m/s(de snelheid van geluid in lucht) te delen. Hierna wordt de tijd teruggemapt(alles tussen de 0 en de 75 wordt terug geschaald naar 0 en 12). Deze waarde wordt voor player 2 4 bits geschoven naar links. En vervolgens wordt het signaal via de hierbovengenoemde SPI verstuurd naar de chip.

4.1. ULTRASONEN

Een unieke uitdaging van ons project is de Ultrasonen besturing, de besturing werkt door middel van een (ultrasonen)sensor aan de rechterzijde van de speler. Deze sensor meet met een speel ruimte van 75cm elke 11 milliseconde. Het aansturen van de ultrasonen sensoren gaat doormiddel van een 2ms lange pulse op de IN-pin. Dit is de trigger van de SRF-04(de door ons gekozen ultrasonen sensor). Hierna verzendt de sensor zijn pulse, op dat moment wordt de OUT-pin ook hoog. Deze blijft hoog tot de gereflecteerde pulse weer binnen komt. Deze tijd wordt gedeeld door 2 omdat het geluid zowel de afstand heen als terug moet afleggen. Deze tijd wordt vervolgens geschaald naar afstand door hem door 29m/s(de snelheid van geluid in lucht) te delen. Hierna wordt de tijd teruggemapt(alles tussen de 0 en de 75 wordt terug geschaald naar 0 en 12). Deze waarde wordt voor player 2 4 bits geschoven naar links. En vervolgens wordt het signaal via de hierbovengenoemde SPI verstuurd naar de chip.

4.2. BUTTONS

Bij de buttons wordt een andere manier van werken gehanteerd, hier wordt de waarde van de plaatsvector onthouden(als integer). En naar gelang welke button geactiveerd wordt, wordt het signaal 1 verhoogd/verlaagd. Dit getal kan maximaal 12 bereiken en minimaal 0. Vervolgens wordt dit signaal voor player 2 ook verschoven, en daarna verzonden via SPI.

De getallen 0->12 voor player 1/2 zijn in gebruik, dit geeft ruimte om 13,14,15 te gebruiken voor andere doeleinde. 13 dient als getal om de Start van systeem aan te geven. 15 is de Reset van het systeem.

5

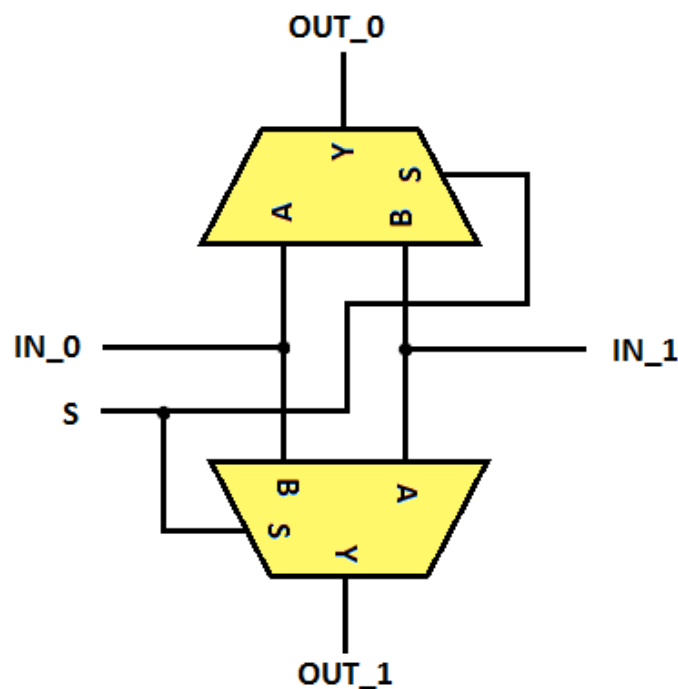
BLACKBOX

5.1. FUNCTIONELE BESCHRIJVING

Voor het genereren van de beelden worden twee SRAM chips gebruikt. De ene chip wordt uitgelezen door de VGA generator en er wordt naar de andere chip geschreven door de GPU. Als een frame geschreven is naar de SRAM chip wordt de chip waar net naar geschreven is verbonden met de VGA generator zodat deze uitgelezen kan worden. De chip die voorheen verbonden was met de VGA wordt doorverbonden met de GPU zodat daar vervolgens naar geschreven kan worden. Door deze techniek toe te passen blijft het beeld op het scherm stabiel. De BlackBox is de schakeling die dat mogelijk maakt.

5.2. INPUTS EN OUTPUTS

De SRAM chips, de VGA en de GPU communiceren met elkaar met behulp van SPI. De schakeling heeft dus twee SPI master aansluitingen voor VGA en de GPU en twee SPI slave aansluitingen voor de twee SRAM chips. Een SPI aansluiting bestaat uit vier signaallijnen. Dit zijn de serial clock (SCLK), master in slave out (MISO), master out slave in (MOSI) en slave select (SS). Ook heeft de schakeling nog een ingang om te selecteren welke SRAM chip verbonden is met de VGA generator en welke met de GPU.



Figuur 5.1: Subschakeling van de blackbox

5.3. IMPLEMENTATIE

De schakeling bestaat uit vier subschakelingen. De subschakeling bestaat uit twee twee naar 1 lijn multiplexers zoals weergegeven in figuur 1. Door de S laag te maken, wordt IN_0 verbonden met OUT_0 en IN_1 met OUT_1. Als S hoog is, is IN_0 verbonden met OUT_1 en vice versa. Vervolgens worden de SCLK lijnen van de GPU en VGA generator verbonden met de ingangen van de subschakeling en de uitgangen met de SCLK lijnen van de twee SRAM chips. Hetzelfde gebeurt ook voor de MOSI en de SS lijnen van de masters en de slaves. Bij de MISO lijnen is het omgekeerd omdat dit een inputlijn is voor de masters en een outputlijn voor de slaves. Hier worden dus de SRAM chips verbonden met de ingangen en de VGA generator en de GPU aan de uitgangen. De BlackBox bestaat nu uit vier subcircuits. Door de select (S) ingangen van de subschakelingen met elkaar te verbinden is de BlackBox nu compleet.

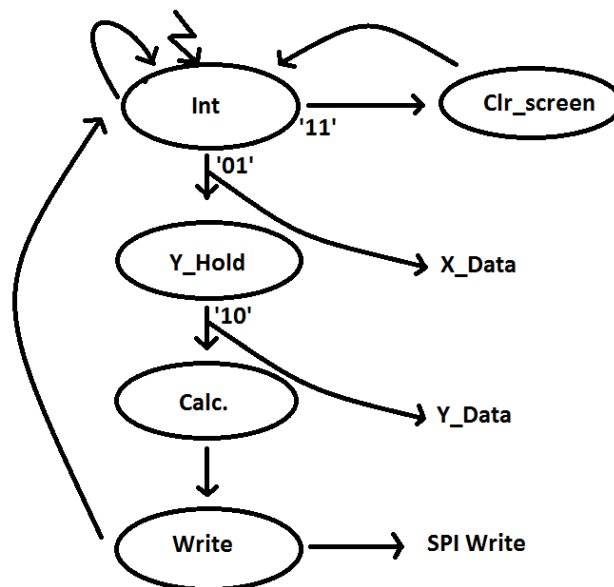
6

ALU/PC

7

GPU

De GPU is ontworpen aan de hand van een FSM, deze bevat 5 states en staat hieronder weergegeven.



Figuur 7.1: Final State Machine van de GPU

In de eerste state kan de GPU drie richtingen op gaan. Als de eerste twee bits van de 8-bits vector '11' zijn gaat de GPU naar Clr-screen, hier zal de VGA zijn scherm leeg maken. De GPU gaat zijn X-data inlezen en springt naar de volgende state (Y-Hold) als de eerste twee bits van de 8-bits vector '01' is. Voor iedere andere waarde op de eerste twee bits van de 8-bits vector blijft de GPU in de int state. Vanuit de Clr-screen state kan de GPU alleen nog maar terug naar de int state en begint de FSM opnieuw. In de Y-Hold state wacht de GPU op Y-Data, de GPU blijft in deze state totdat de eerste twee bits van de 8-bits vector '10' aangeven (Duurt 20 klokslagen). Op dat moment leest de GPU de Y-Data in en stapt over naar de Calc. state. In deze state wordt het adres berekend voor de VGA, dit gebeurt met de formule $32 * Y + X = \text{adres}$. Als deze berekening voor het bepalen van zijn adres is voltooid springt de GPU automatisch naar de volgende state, de Write state. In de Write state schrijft de GPU data uit de buffers weg via SPI. Automatisch zal de GPU terug springen naar zijn eerste state, het proces begint dan opnieuw.

8

VGA

9

SPI

Serial Peripheral Interface in het kort ook SPI genoemd is een de facto-standaard die ooit door Motorola bedacht is voor het communiceren met randapparatuur, zoals embedded systems, sensors en memory cards. Bij SPI heb je de master en de slave deze communiceren met elkaar via twee seriële datalijnen daarnaast is er nog een kloklijn die aangeeft wanneer er data overgedragen wordt. De kloklijn wordt aangestuurd door de master, dit houdt in dat de slave niet kan pauzeren en altijd data aan de master aan moet bieden. De standaard definieert enkel hoe de data van de master naar de slave en vice versa over wordt gedragen, niet hoe de data verwerkt wordt, dit heeft als gevolg dat randapparatuur zelf nog moet definiëren hoe de communicatie verloopt.

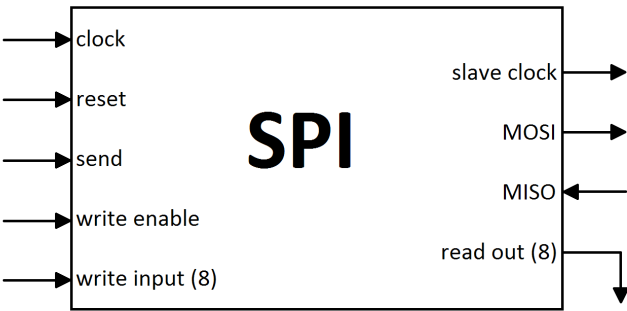
9.1. FUNCTIONELE BESCHRIJVING

De SPI module begint met zenden/ontvangen zodra het send signaal hoog is en stopt zodra er acht bits zijn verzonden/ontvangen, daarna kunnen de ontvangen bits uitgelezen worden en nieuwe bits in worden geladen om te verzenden. Het overbrengen van het shift register dat in de master zit naar de slave en vice versa deze shift registers zijn in een kring op elkaar aangesloten, waarbij altijd de meest significante bit wordt verzonden. Voor de communicatie met de slave wordt een slave klok signaal gegenereerd, de data overdracht is gesynchroniseerd ten opzicht van dit signaal. Data wordt gesampled op de opgaande klokflank en de data wordt geshift op de neergaande klokflank.

9.2. INPUTS EN OUTPUTS

Hieronder een beschrijving van alle in- en outputs van de SPI module zoals te zien in figuur 9.1.

- Clock: het kloksignaal waarop de SPI draait, het slave clock signaal zal dezelfde frequency hebben als dit kloksignaal
- Reset: de hoofd reset van de SPI
- Send: de input die aangeeft wanneer er begonnen moet worden met zenden/ontvangen
- Write_enable: als deze hoog is zal write_in(8) in het shift register geladen worden
- Write_in(8): de bits die naar het shift register worden geschreven als write_enable hoog is
- Read_out(8): de waarde die in het shift register staat
- Slave clock: het slave kloksignaal die de communicatie met de slave aanstuurt
- MOSI: de datalijn van de master naar de slave
- MISO: de datalijn van de slave naar de master



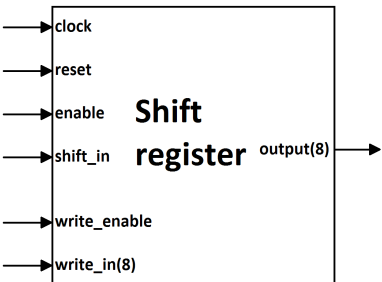
Figuur 9.1: Diagram of SPI circuit

9.3. IMPLEMENTATIE

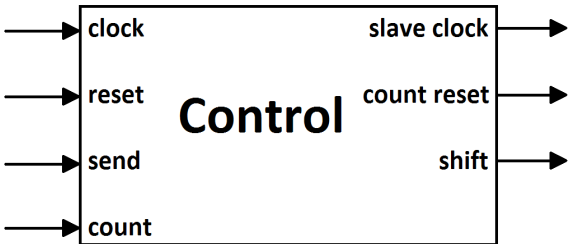
Voor de implementatie van de SPI is er voor gekozen om het in drie subsystemen op te delen:

- Counter: een simpele teller die de opgaande klokslagen van het slave clock signaal telt.
- Shift register: een shift register van acht bits die shift op de neergaande klokflank als het enable signaal hoog is en nieuwe waardes inlaad als de write enable hoog is. Het blokschema van het shift register is te zien in figuur 9.2.
- Control: een statemachine die er voor zorgt dat de SPI stopt met shiften na 8 klokslagen van de slave klok, zodat er tijd is om het register uit te lezen of nieuwe waarden in te laden. Het blokschema van Control is te zien in figuur 9.3.

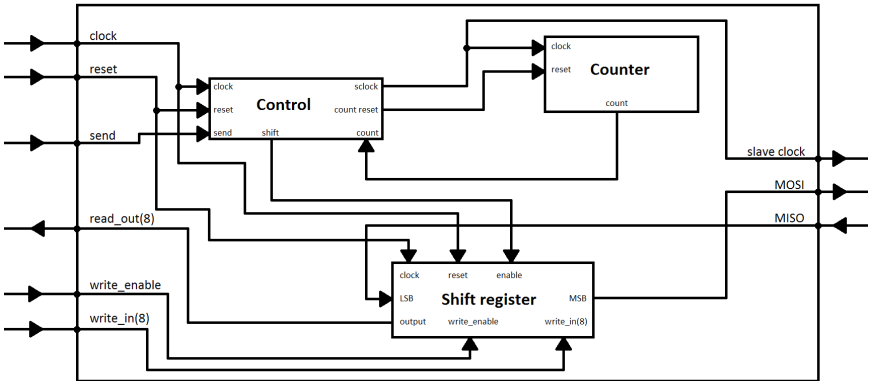
Deze drie subsystemen zijn aan elkaar verbonden volgens het schema in figuur 9.4.



Figuur 9.2: Diagram van Shift register



Figuur 9.3: Diagram van Control



Figuur 9.4: Diagram van de verbinding van de componenten

10

SD-KAART

Voor het opslaan van de instructies voor de CPU is gekozen voor een SD-kaart. Voor het uitlezen van een SD-kaart zijn er drie verschillende modes, twee hiervan zijn echter gebaseerd op een zelf ontworpen systeem van SanDisk. De derde mode is echter gebaseerd op SPI waarvan al een module is die al ontworpen is, daarom is hier ook voor gekozen.

10.1. SD COMMUNICATIE

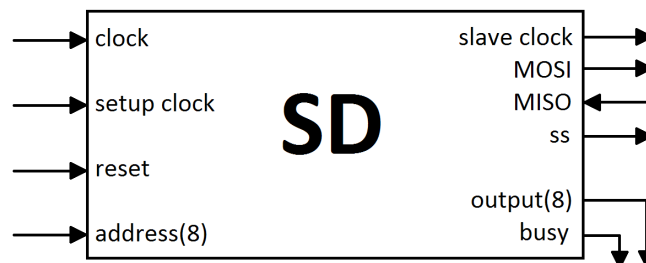
Aangezien SPI niets zegt over hoe de communicatie verloopt of wat de betekenis van de overgebrachte data is, definieert SanDisk in de SD specificatie een aantal commando's die naar de SD-kaart gestuurd kunnen worden. Er zal niet specifiek in worden gegaan op wat deze commando's zijn, je moet denken aan: reset, lezen en schrijven commando's, er zal wel ingegaan worden op hoe deze commando's worden gestuurd en wat de reacties hier op zijn.

10.2. SD COMMANDO'S & RESPONSES

De communicatie met de SD-kaart verloopt als volgt: er wordt een commando naar de SD-kaart verstuurd, de SD-kaart geeft antwoordt of het een correct commando is en of het commando verwerkt kan worden, daarna komt eventueel nog, bij het lezen, data van de SD-kaart of, bij het schrijven, data naar de SD-kaart. Zodra alles is afgehandeld kan er een nieuw commando gestuurd worden.

10.3. FUNCTIONELE OMSCHRIJVING

De SD module is zo ontworpen dat deze een adres binnen krijgt van de program counter van de CPU deze uitleest vanaf de SD-kaart en deze vervolgens aanbiedt aan de CPU. Zolang de SD module nog bezig is met het uitlezen van het adres zal het busy signaal hoog zijn.



Figuur 10.1: Diagram of SD circuit

10.4. INPUTS EN OUTPUTS

Hieronder een beschrijving van alle in- en outputs van de SD module zoals te zien in figuur 10.1.

- Clock: het kloksignaal waar de SD module op draait, dit is ook het kloksignaal waar de SPI na initialisatie op zal draaien
- Setup clock: het kloksignaal dat tijdens de initialisatie gebruikt zal worden, deze moet tussen de 100 en 400 kHz liggen
- Reset: het algemene reset signaal
- Address: het adres dat van de SD-kaart zal worden afgelezen
- Output(8): dit zal de waarde zijn die op het adres staat zodra deze is uitgelezen
- Busy: dit signaal geeft aan of de SD module bezig is
- Slave clock: het slave kloksignaal dat de SD-kaart aanstuurd
- MOSI: de datalijn van de SD module naar de SD-kaart
- MISO: de datalijn van de SD-kaart naar de SD module
- SS: het slave select signaal, dit signaal wordt gebruikt om de SD-kaart in SPI mode te krijgen en om aan te geven dat de SD-kaart geselecteerd is.

11

GAME: PONG

12

DISCUSSIE

Over het eindontwerp is nog geen discussie mogelijk. Voor het schrijven van dit tussentijdse ontwerprapport zijn er nog wel een aantal dingen naar voren gekomen. Voorop staat het feit dat alles minder snel en soepel verloopt als dat we gehoopt hadden. Het testen van de hardware op het FPGA bord bracht veel oponthoud met zich mee. Ook bleken de stappen die tot het eindproduct moesten lijden niet compleet bedacht, tijdens het ontwerpproces kwamen er extra stappen naar voren die veel tijd hebben gekost. Ondanks dat de einddatum steeds dichterbij komt denken we dat alles binnen de tijd realiseerbaar is.