

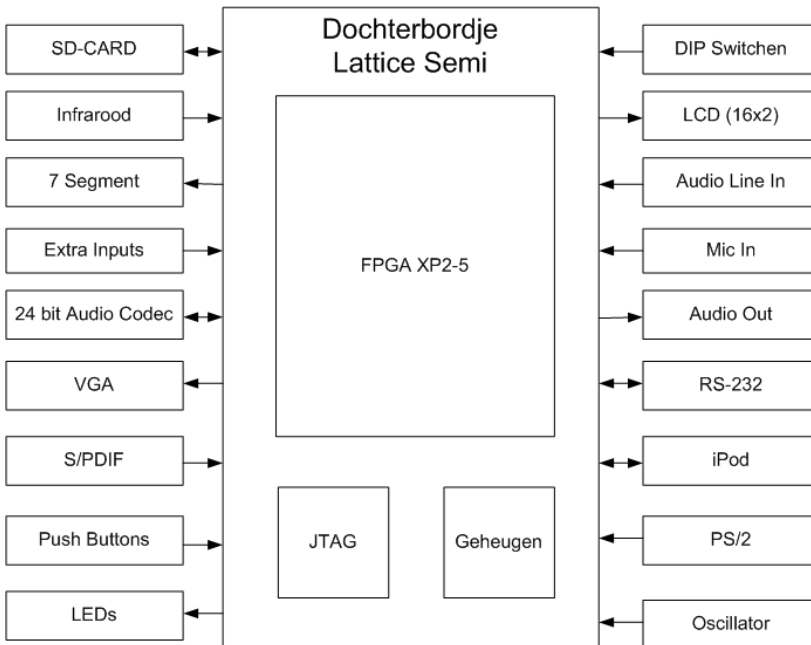
<b>14.</b>	<b>Hoofdstuk 14 Lattice development board LFXP2-5E .....</b>	<b>3</b>
14.1	Functioneel overzicht development board .....	3
14.2	Dochterbordje .....	5
14.3	LFXP2-5E (FPGA van Lattice).....	7
14.3.1	Programmeren .....	9
14.4	Voeding .....	11
14.5	Eenvoudige interfaces .....	12
14.5.1	Leds .....	12
14.5.2	Push buttons.....	12
14.5.3	Dip switches .....	13
14.5.4	Oscillator socket .....	13
14.5.5	Extra digitale inputs .....	14
14.6	Gemultiplexeerde 7-segment displays .....	15
14.7	PS/2 Interface .....	17
14.8	LCD .....	19
14.8.1	Interface .....	19
14.8.3	Geheugens en registers.....	20
14.8.4	commando's .....	22
14.8.5	De initialisatie van het LCD .....	24
14.8.6	De werking .....	25
14.9	RS-232 .....	26
14.9.1	Inleiding.....	26
14.9.2	Interface .....	26
14.9.3	Protocol .....	28
14.10	VGA .....	29
14.10.1	Inleiding.....	29
14.10.2	Interface .....	29
14.10.3	Protocol .....	30
14.11	Infrarood .....	34
14.11.1	Inleiding.....	34
14.11.2	Interface .....	34
14.11.3	Protocol RC-5 .....	35

14.12	S/Pdif .....	36
14.12.1	Inleiding .....	36
14.12.2	Interface .....	36
14.13	Audio gedeelte .....	37
14.13.1	Opbouw .....	37
14.13.2	Line inputs .....	39
14.13.3	Microfoon ingang .....	40
14.13.4	Output .....	41
14.13.5	Audio codec .....	41
14.13.6	Software controle interface .....	42
14.13.7	ADC en DAC .....	43
14.13.8	Audio interface .....	44
14.13.9	Modes .....	45
14.13.10	Registers .....	45
14.14	iPod interface .....	46
14.14.1	Inleiding .....	46
14.14.2	Interface .....	46
14.14.3	Protocol .....	47
14.15	SD-card .....	50
14.15.1	Inleiding .....	50
14.15.2	Interface .....	51
14.15.3	Protocol .....	52
14.16	FPGA .....	53
14.17	Overzicht van de I/O pinnummering van de FPGA XP2-5 .....	55
14.17.1	Verduidelijking van de kolommen: .....	57

## 14. Hoofdstuk 14

### Lattice development board LFXP2-5E

#### 14.1 Functioneel overzicht development board.



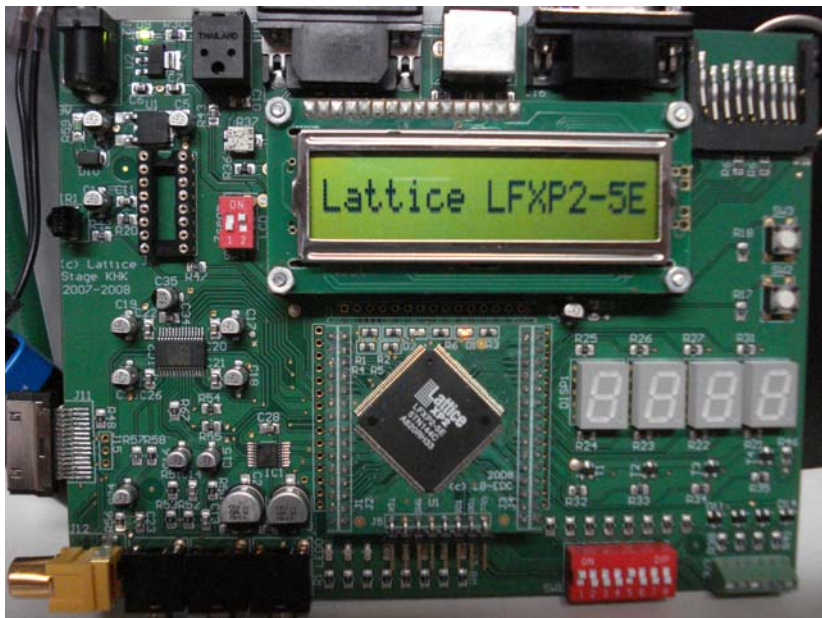
Figuur 1 functioneel overzicht development board

Figuur 1 toont het functioneel overzicht van het development board. Centraal is hier het dochterbordje van Lattice Semiconductor dat later in dit hoofdstuk besproken wordt. Hierop bevindt zich de FPGA. De FPGA is het hart van het development board en bestuurt alle interfaces. Deze interfaces worden voorgesteld door de verschillende blokken rond het dochterbordje. Met de pijlen wordt aangegeven in welke richting de communicatie tussen de interface en de FPGA verloopt.

Een korte omschrijving van de verschillende interfaces op het bordje:

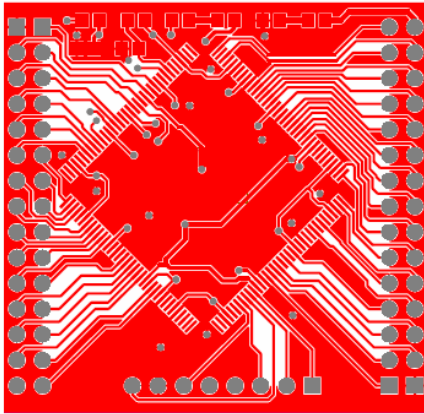
1. LEDs: 8 ledjes.
2. Push buttons: 2 eenvoudige drukknoppen.
3. Dip switches: De 8 dipswitches kunnen gebruikt worden om een byte voor te stellen.
4. Oscillator socket: Hier kan een oscillator ingeplugd worden indien we een ander kloksignaal voor de FPGA willen gebruiken.
5. Extra inputs: Dit zijn 4 extra ingangen waar men bijvoorbeeld een extern kloksignaal kan aanhangen.
6. 7 segment: Vier 7-segment displays. Deze worden gemultiplexeerd aangestuurd.

7. PS/2: Dit biedt de mogelijkheid om een toetsenbord of muis in te lezen.
8. LCD: Een LCD aansluiting is voorzien voor het uitlezen van allerlei gegevens.
9. RS-232: Deze aansluiting laat communicatie toe met een computer.
10. VGA: Voor het aansluiten van een computerscherm.
11. Infrarood: Een IR-ontvanger. Hiermee kunnen we bijvoorbeeld via een afstandsbediening de FPGA allerlei acties laten uitvoeren.
12. S/Pdif: Om de audio afkomstig van de optische uitgang van een CD- of DVD-speler in te lezen.
13. 24-bit audio codec: Hoort bij Audio Line in, mic in en audio out. Deze bezit een analoog naar digitaal convertor (ADC) en een digitaal naar analoog convertor (DAC). De codec wordt bestuurd door de FPGA.
14. Mic in: Aansluiting voor een microfoon.
15. Audio line in: Hier kan men een audiosignaal aansluiten van bijvoorbeeld een MP3 speler.
16. Audio out: De uitgang van het gehele audio gedeelte.
17. iPod: Er is ook een iPod connector voorzien. Zo kan men bijvoorbeeld een iPod dock simuleren. De IR ontvanger wordt dan gebruikt om een afstandsbediening in te lezen. De codec en audio uitgang worden gebruikt om geluidboxen aan te sluiten.
18. SD-card: Een connector voor het inpluggen van een SD-card. Dit biedt uitgebreide mogelijkheden zoals gegevensopslag en het inlezen van gegevens.



Figuur 2 Lattice XP2 development board

## 14.2 Dochterbordje

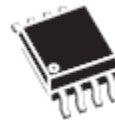
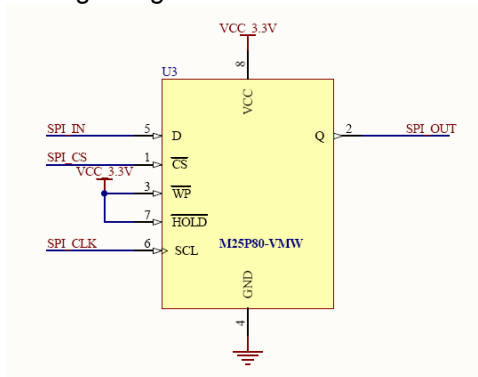


Figuur 3 Dochterbordje

Zoals vermeld is het dochterbordje dus een opsteek-printje. Het bordje bevat een XP2-5 FPGA van Lattice.

Het is een 60 pins printje met een FPGA, een 1,2V voedingsregelaar voor de kern van de FPGA en een JTAG interface. Ook is er de mogelijkheid voorzien hier een SPIFlash geheugen-chip (8pin SOIC uitvoering) te plaatsen.

Extra geheugen



Figuur 4 8pin SOIC SPIFlash geheugen in schema dochterbordje.

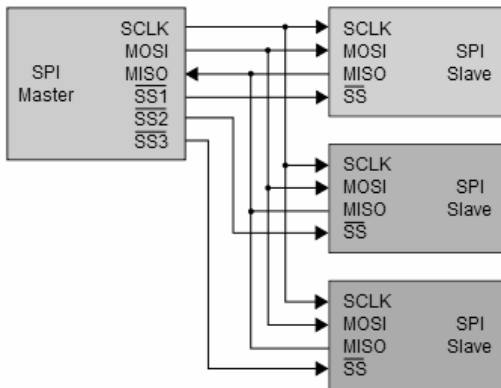
Het dochterbordje kan dus uitgebreid worden met extra flash-geheugen. Dit moet wel in de vorm van een 8pin SOIC (Small-Outline Integrated Circuit). Verder kan het SPIFlash geheugen van eender welk merk en type zijn, zolang het IC maar compatibel met de SPIFlash standaard is.

Een SPIFlash geheugen is allereerst niet-vluchtig geheugen en heeft een seriële SPI (Serial Peripheral Interface) bus interface. Ze zijn beschikbaar van 512Kbit tot 128Mbit. De frequentie waarmee ze werken ligt in de groot-

teorde van tientallen MHz. Het geheugen is opgebouwd in sectoren van elk een aantal pagina's. Elke pagina is dan een aantal bytes breed. Het aansturen van een serieel Flash geheugen is eenvoudig omdat er eenvoudige instructies voor lezen, wissen en programmeren gebruikt worden. Het IC wordt dus aangesproken in SPI mode. Dit is een seriële data link standaard die full duplex communicatie gebruikt. De IC's werken in een master/slave mode. In ons geval is de FPGA dus de master. Meestal wordt deze standaard gebruikt om meerdere slave IC's te laten communiceren met één master IC. Hiervoor bezitten de slaves een "chip select ingang". Hiermee selecteert de master dus de IC waar hij mee wil communiceren. Voor de communicatie worden 4 signalen gebruikt:

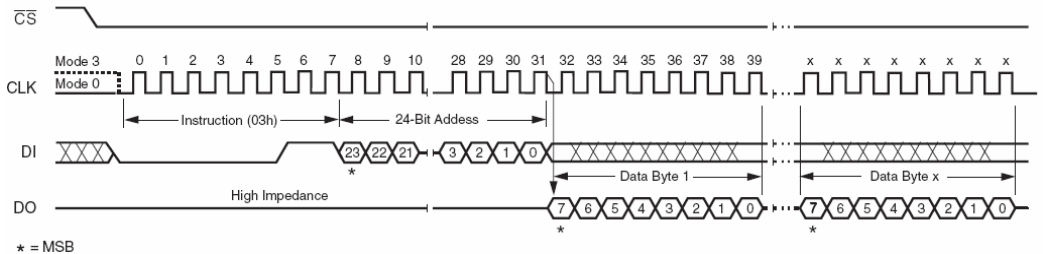
- De seriële klok die door de FPGA gegenereerd wordt. Hier SPI\_CLK
- De master out, slave input data lijn (MOSI). Hier SPI\_IN
- De master in, slave output data lijn (MISO) Hier SPI\_OUT
- De slave select. (Of "Chip select", enkel gebruikt bij meerdere slaves). Hier SPI\_CS

/WP uit het schema staat voor write protect en /hold om de data overdracht te pauseren maar omdat dit niet nodig is worden ze beiden aan Vcc gehangen.



Figuur 5 SPI

In figuur 5 wordt een voorbeeld van een dataoverdracht bij SPIFlash voorgesteld. Dit stelt een leesinstructie voor. Wanneer de chip select laag wordt gemaakt (= actief), kan een leesinstructie byte (03hex) naar het flash geheugen verzonden worden. Dit wordt ingeklokt gevolgd door een 24-bit adres. Dit adres bevat de plaats waar de gewenste data zich bevindt. De data kan nu continu uit het flash geheugen geklokt worden tot de chip select terug hoog wordt gemaakt.



Figuur 6 Leesoverdracht

### 14.3 LFXP2-5E (FPGA van Lattice)

Het dochterbordje bevat een XP2-5 FPGA van Lattice. Deze FPGA is uitgevoerd in een 144 pins TQFP behuizing. XP2 duidt hier op de familienaam van de FPGA. Terwijl 5 duidt op 5k Lookup Tables, dus 5000 LUTs.

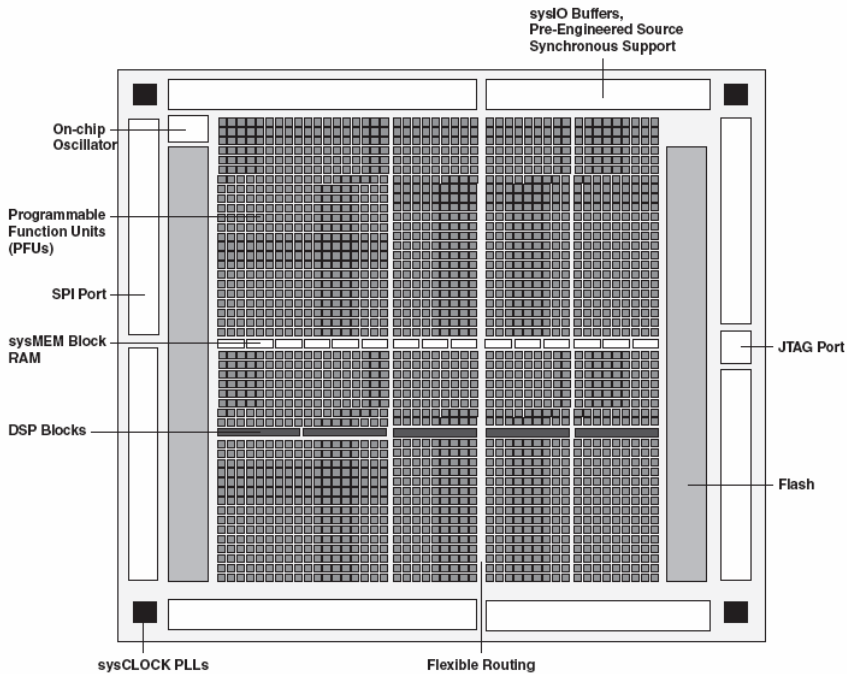


Figuur 7 Lattice XP2 FPGA

Tabel 1 Lattice XP2 familie

Device	XP2-5	XP2-8	XP2-17	XP2-30	XP2-40
LUTs (K)	5	8	17	29	40
Distributed RAM (KBits)	10	18	35	56	83
EBR SRAM (KBits)	166	221	276	387	885
EBR SRAM Blocks	9	12	15	21	48
sysDSP Blocks	3	4	5	7	8
18 x 18 Multipliers	12	16	20	28	32
V <sub>CC</sub> Voltage	1.2	1.2	1.2	1.2	1.2
GPLL	2	2	4	4	4
Max Available I/O	172	201	358	472	540
<b>Packages and I/O Combinations</b>					
132-Ball csBGA (8 x 8 mm)	86	86			
144-Pin TQFP (20 x 20 mm)	100	100			
208-Pin PQFP (28 x 28 mm)	146	146	146		
256-Ball ftBGA (17 x 17 mm)	172	201	201	201	
484-Ball fpBGA (23 x 23 mm)			358	363	363
672-Ball fpBGA (27 x 27 mm)				472	540

De XP2 bevat non-volatile flash geheugen. Bij het aanzetten van de voedingsspanning worden hieruit de configuratiedata gehaald en getransfereerd naar de configureerbare SRAM. Dit gebeurt in microseconden. Dit noemt men de "Instant-on" technologie.



Figuur 8 Architectuur XP2

Figuur 8 toont de architectuur van een XP2-FPGA. Hieronder worden de verschillende onderdelen kort besproken.

**PFU:** De kern van de XP2 bestaat uit logische blokken; PFU's genaamd. Ze kunnen geprogrammeerd worden als logische functies.

**GPLL:** De FPGA bevat 2 GPLL's (General Purpose Phase Locked Loop). Een GPLL wordt gebruikt om een frequentie te synchroniseren met een andere of om eender welke frequentie te genereren vertrekkend van een ingangsfrequentie. Dit wordt gedaan met een PLL en frequentiedelers.

**sysMem Block Ram:** Een EBR (embedded block ram) bestaat uit 18kBit SRAM geheugen met specifieke in- en uitgangsregisters.

**sysDSP blocks:** Speciale blocks gemaakt voor DSP (digital signal processing) toepassingen. Dit houdt dus in dat ze ontworpen zijn voor het maken van complexe logische functies.

**On chip oscillator:** Een interne CMOS oscillator. 2.5 – 163 Mhz selecteerbaar.

**JTAG en SPI port:** Deze kunnen gebruikt worden om het Flash geheugen te programmeren.



**SYSIO:** Elke PIC (programmable I/O cell) bevat 2 PIO's (programmable I/O's). Deze hebben een buffer. Ze vallen allemaal onder een sysIO bank. De XP2 FPGA's hebben zo 8 banken. Elke bank kan gebruik maken van een aparte I/O voedingsspanning VCCIO. De banken kunnen geconfigureerd worden voor verschillende standaard I/O systemen zoals: LVTTTL, LVCMOS33, LVCMOS25, LVCMOS12, LVDS, HSTL en SSTL.

#### 14.3.1 Programmeren

Voor het programmeren van een FPGA moet er allereerst een ontwerp inge-  
ven worden. Dit kan in de vorm van een schema, een gewenste golfvorm of  
een geschreven programmacode (VHDL, verilog, ...).

VHDL (VHSIC Hardware Description Language) is de standaard in de indu-  
strie voor de beschrijving van een digitale schakeling. VHDL is niet alleen  
een beschrijving van een systeem maar bevat ook de oplossing. Deze is dan  
echter wel op een zodanig hoge manier beschreven dat de computer zelf  
nog heel wat werk heeft om daar een werkende schakeling voor samen te  
stellen.

Voor het ingeven van een VHDL ontwerp werkt men met een "logic design  
tool". Hierin zit normaal ook een fitter. Deze minimaliseert het ontwerp en  
bevat ook de software voor compilatie. Bij compilatie wordt het design ver-  
werkt en omgezet naar de eigenlijke code die in de component moet worden  
binnengeschoven. Het resultaat hiervan is een JEDEC file.

De informatie over welke interne verbindingen er gemaakt of verbroken  
moeten worden zit dus bevat in een JEDEC (Joint Electron Device Enginee-  
ring Council)-bestand. Dit JEDEC-bestand wordt door de programmeersoft-  
ware ingelezen en doorgegeven aan de programmeerhardware die er ver-  
volgens voor zorgt dat de juiste verbindingen in het IC gemaakt worden.

Tegenwoordig kan men de programmeerbare IC programmeren terwijl hij in  
zijn werkomgeving zit. Bijvoorbeeld op een PCB. Deze technologie noemt  
met ISP. ISP staat voor "In System Programmability".

Als programmeerprotocol wordt veelal het JTAG-protocol gebruikt. Het JTAG  
(Join Test Ation Group) protocol IEEE 1149.1 heeft als doel het programme-  
ren, het testen en debuggen van IC's in de schakeling (in circuit). De softwa-  
re en de componenten hebben allemaal een JTAG-kern die garant staat  
voor een transparante communicatie.

Een JTAG interface is een 4 of 5-pins interface die toegevoegd is aan een chip:

1. TDI (Test Data In)
2. TDO (Test Data Out)
3. TCK (Test Clock)
4. TMS (Test Mode Select)
5. TRST (Test Reset) optional.

JTAG is een serieel protocol. Één bit wordt per TCK clockimpuls in TDI en uit TDO getransfereerd. De TCK frequentie varieert afhankelijk van de chip maar is normaal iets in de grootteorde van 10 tot 100Mhz. De configuratiemode wordt bepaald door de TMS-pin. De TRST pin is niet altijd voorzien. Dit is een RESET-pin. Wanneer deze niet voorzien is wordt een reset gewoon via TDI synchroon ingeklokt.

Voor het programmeren van het development board wordt gebruik gemaakt van de ispDOWNLOAD kabel van Lattice. Wanneer men klaar is met het logisch ontwerp (VHDL, Verilog) in ispLEVER maakt men de programming file (JEDEC) voor de FPGA. Tenslotte moet de programming file verzonden worden naar het device en dit met de software ispVM van lattice.

## 14.4 Voeding

Een voeding van 3,3V is voorzien voor de FPGA.

Deze 5V voorziening is nodig voor het voeden van de LCD, de PS/2 interface, en de 7-segment displays.

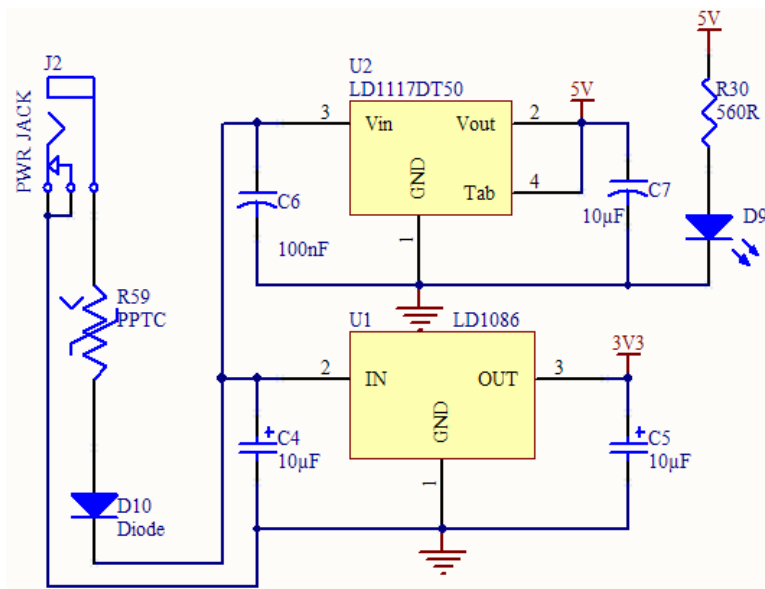
De voeding is opgebouwd rond 2 spanningsregelaars van STMicroelectronics. De LD1117 voorziet 3,3V en de LD1086 voorziet 5V.

De LD1117 is een low dropout voltage regulator (typisch 1V) die een stroom kan leveren tot 800mA en 3V3 levert met een tolerantie van 1% bij 25°C.

Enkel een normale 10µF condensator is nodig aan de uitgang voor stabiliteit.

De LD1086 is tevens een low dropout voltage regulator (typisch 1.3V) en kan een output stroom leveren tot 1.5 A en levert 5V met dezelfde tolerantie.

Zoals in de datasheets aangegeven zijn elco's van 10µF nodig.



Figuur 9 Voeding

De spanning wordt dus aangesloten aan powerjack J2 met een adapter van ongeveer 9V.

Een polyswitch R59 is opgenomen in het circuit ter beveiliging tegen overbelasting en kortsluiting. Dit is een soort van weerstand die, wanneer een te hoge stroom dreigt te vloeien, een oneindig hoge weerstand aanneemt zodat de kring onderbroken blijft. Deze polyswitch treedt na herstelling van de fout terug naar zijn oorspronkelijke weerstandswaarde.

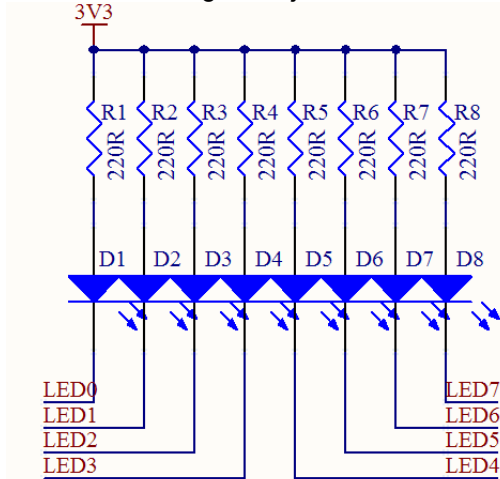
Er is ook een diode D10 voorzien achter de polyswitch. Dit is een ompooldiode die de voeding beschermt tegen het aansluiten van verkeerd gepolariseerde spanningen.

Aan de 5V uitgang is ook een indicatieled voorzien die oplicht wanneer de voeding actief is.

## 14.5 Eenvoudige interfaces

### 14.5.1 Leds

Zoals op de meeste development boards zijn ook op dit bordje enkele leds voorzien. In ons geval zijn er acht leds voorzien. Ze zijn actief laag.



Figuur 10 Leds

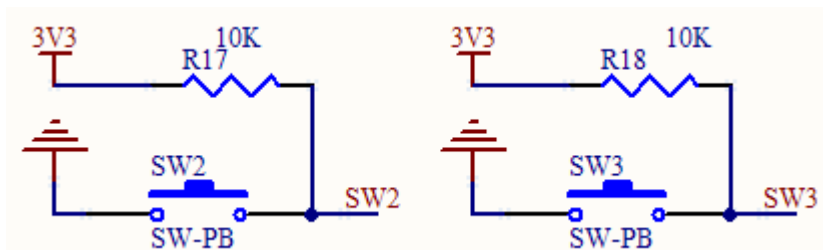
De leds zijn hier HSMG-C120 leds van Agilent Technologies.

Er is gekozen voor een stroom van 6mA waaruit een weerstand van 220 ohm wordt bekomen bij een spanningsval van 2V over de led.

Bij 6mA krijgen we een voldoende grote lichtopbrengst.

### 14.5.2 Push buttons

De push buttons zijn ook actief laag gekozen. Ze worden dus geschakeld naar massa en indien niet actief verbonden met een pull-up weerstand van 10kohm naar 3V3 voeding.



Figuur 11 Push buttons

Hierbij moeten wel opgemerkt worden dat bij bedienen van een push button er geen stabiel signaal veroorzaakt wordt. Er ontstaat dender. Bij het bedienen van de drukknop komen er als het ware meerdere stijgende en dalende flanken voor op de uitgang van de drukknop. Na ongeveer 20ms is dit sig-

naal wel stabiel. Wil men dus softwarematig werken met flankdetectie van de drukknop dan zal men hiervoor een stukje “ontdender-code” voor moeten schrijven waar men gebruik maakt van een schuifregister.

**Bijvoorbeeld:**

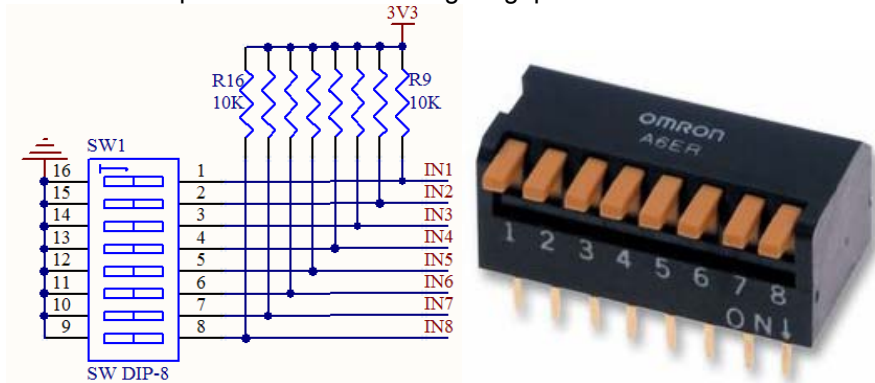
```

WAIT UNTIL (Clock_50Hz<EVENT) AND (Clock_50Hz = '1');
state(3 downto 1) <= state(2 downto 0);
state(0) <= button;
IF( state = "0000") THEN
    hit <= '1';
ELSE hit <= '0';
END IF;

```

**14.5.3 Dip switches**

Ook voor de dip switches is actief laag toegepast.

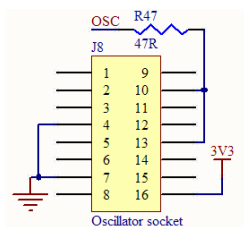


Figuur 12 Dip switches

Door het gebruik van 8 dipswitches kunnen we een bepaalde BCD waarde voorstellen op een gemultiplexeerd displayaansturing. Ook hier is gekozen voor pull-up weerstanden van 10K.

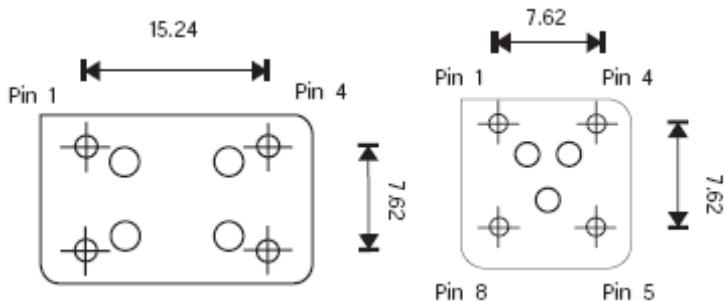
**14.5.4 Oscillator socket**

Op het bordje is ook een socket voorzien die toelaat een externe crystal oscillator in te pluggen. Hierdoor is het development board flexibel in het gebruik van frequenties. Dit kloksignaal komt binnen op een speciale PCLK pin (Primary clock) van de FPGA. Er wordt gebruik gemaakt van DIL (Dual In Line) klok oscillatoren.



Figuur 13 Oscillator socket

De connector is een gewone DIL-16 of DIL-14 IC socket connector met een pitch van 2.54mm.



### Pin Connections

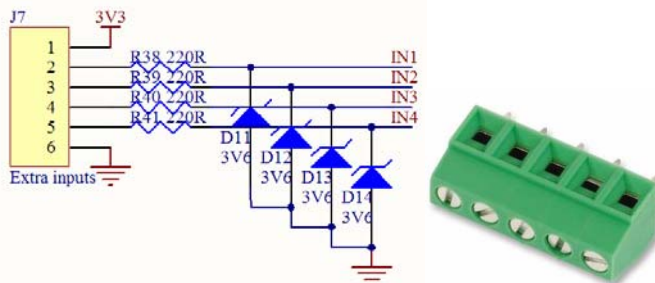
- Pin 1 - N.C (Enable/Disable)
- Pin 4 - Ground
- Pin 5 - Output
- Pin 8 - +V DC

Figuur 14 Onderaanzicht DIL oscillatoren

Figuur 14 toont voorbeelden van “dual-in-line clock oscillatoren”. Links is een 14-pin DIP(Dual In line Pin) afgebeeld en rechts een 8-pin DIP.

### 14.5.5 Extra digitale inputs

Op het bordje zijn er 4 extra inputs voorzien. Deze kunnen gebruikt worden als ingang voor een frequentiemeter. Dit gebeurt in de vorm van een 6-way terminal block. Vier pinnen zijn hiervan inputs waarvan IN1 met een GPLLC ingang (General purpose PLL input pad) van de FPGA verbonden is. Met de andere 2 wordt er 3V3 en GND aangeboden om een externe toepassing van spanning te voorzien.

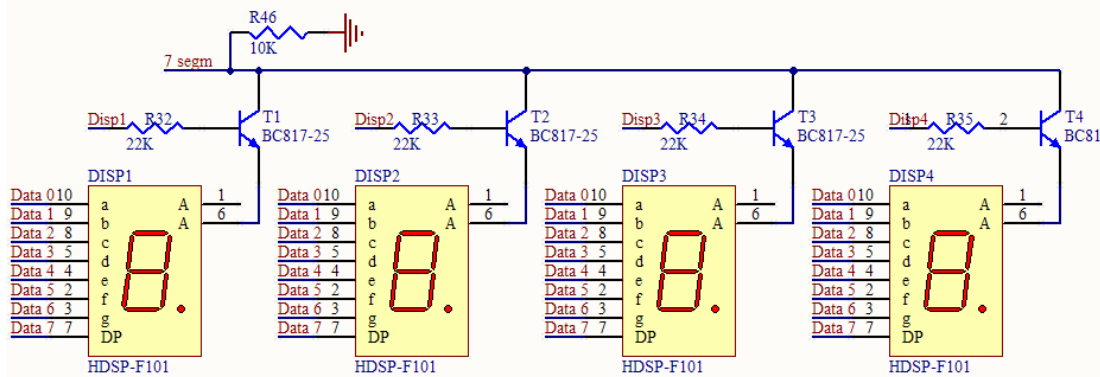


Figuur 15 Extra digitale ingangen

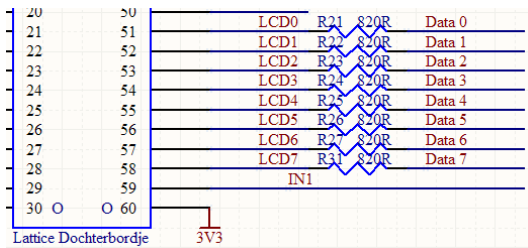
Bij deze inputs is een beveiliging voorzien voor het aansluiten van 5V signalen. Dit is gerealiseerd door een weerstand – zenerdiode netwerkje. De zenerdiodes zijn van het type 3.6V omdat de inputs van de FPGA max. 3.75V kunnen accepteren.

## 14.6 Gemultiplexeerde 7-segment displays

Op het development board zijn vier 7-segment displays voorzien. Ze worden gemultiplexeerd aangestuurd. Dit houdt in dat de displays één voor één aangestuurd moeten worden. Dit is zo gedaan omdat dit het aantal benodigde I/O's voor deze interface vermindert. In dit geval hebben we maar 11 lijnen nodig (7 voor data en 4 voor selectie van de displays). Zonder gemultiplexeerde aansturing zouden er 28 lijnen nodig zijn.



Figuur 16 Gemultiplexeerde displayaansturing



Figuur 17 de voorschakelweerstand

De displays worden dus één voor één voorzien van voeding door de schakeltransistoren T1 tot en met T4. De acht datalijnen hangen parallel aan elke display. De displays moeten geschakeld worden op een frequentie van +- 200 Hz om flikkering te voorkomen. Zo is de frequentie voor elk display 50hz.

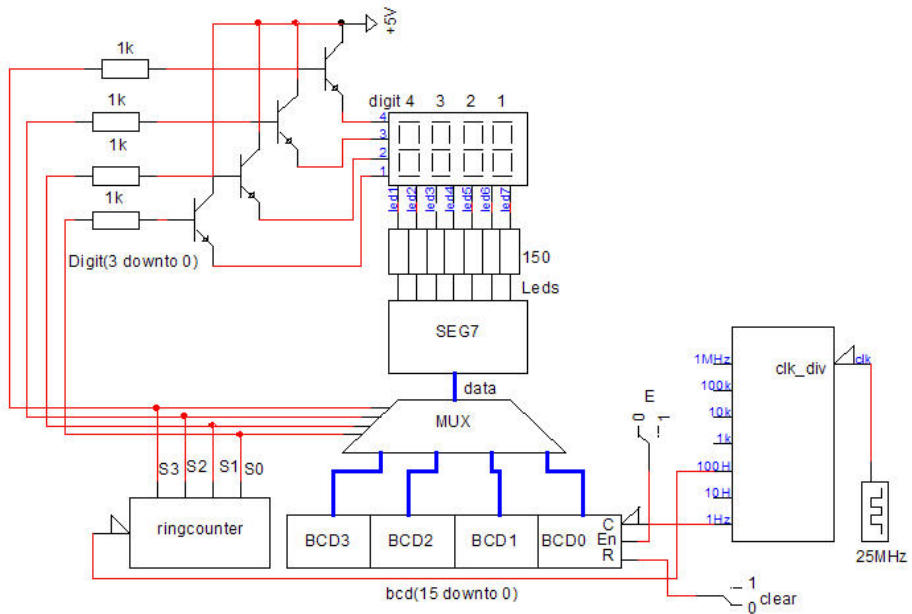
Wanneer er nu van display verandert wordt moeten ook andere data op de data-uitgangen gestuurd komen. Dit alles geeft een effect van 4 displays die continu aan zijn omwille van de traagheid van het oog.

Dit principe wordt nog eens verduidelijkt met onderstaande afbeelding.

De ringteller zit dus geprogrammeerd in de FPGA en zorgt voor een logische '1' op de basis van de transistor.

De data die op het 7 segment getoond moet worden moet samen met de gewenste display actief zijn zodat het juiste cijfer op het juist display verschijnt.

Het onderste gedeelte wordt geprogrammeerd in de FPGA. Dus de ringteller, Multiplexer en 7segment decoder.



De 7-segment displays zijn van het type “Common Anode”. Dit wil zeggen dat de datalijnen aangestuurd worden met een logische ‘0’. Het zijn HDSP-F101 displays van de fabrikant Avago Technologies en worden gebruikt vanaf een  $I_F$  van 1mA.

Wij hebben gekozen voor een stroom van 2mA wat een voldoende lichtopbrengst geeft. Hierdoor verkrijgen we een spanningsval van 1,7V wat leidt tot voorschakelweerstand van 820ohm.

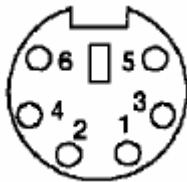
De totale stroom die er dus getrokken kan worden (wanneer alle segmenten aan zijn) bedraagt dus  $8 \times 2 \text{ mA} = 16\text{mA}$ .

De NPN schakeltransistor BC817-16 van Philips heeft een  $h_{FE}$  van ongeveer 150. Hieruit verkrijgen we een  $I_B = I_C / h_{FE} = 0,106\text{mA}$  waaruit  $R_B = (V_{CC} - V_{BE}) / I_B = (3,3\text{V} - 0,6\text{V}) / 0,106\text{mA} = \text{ongeveer } 25\text{K}$  waaruit een  $R_B = 22\text{K}$  gekozen wordt zodat deze zeker schakelt.



## 14.7 PS/2 Interface

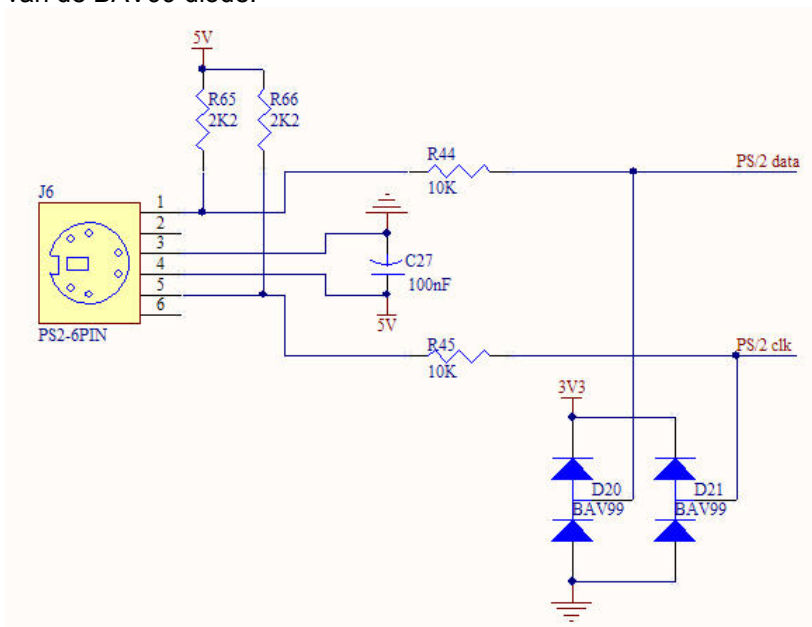
Een PS/2 connector bestaat uit 6 pinnen:



1	Data pin	4	VDD
2	NC	5	Clock pin
3	Ground	6	NC

Figuur 18 PS/2 interface

De PS/2 interface wordt gebruikt om een toetsenbord in te lezen. We gaan er vanuit dat we alleen van het toetsenbord naar FPGA toe data gaan verzenden en niet omgekeerd. De interface bestaat uit 2 lijnen: een kloklijn die gecontroleerd wordt door het toetsenbord en een datalijn. We hebben PS/2 gevoed met 5V omdat we hier ook de mogelijkheid willen bieden om in de toekomst een muis als besturingselement te gebruiken en dit niet werkt op 3,3V. Omdat de interface gevoed is op 5V zijn ook de signalen die naar de FPGA gestuurd worden 5V-signalen. Deze zijn echter weer schadelijk voor de FPGA omdat die maximum 3,75V aanvaardt als inputsignaal. Daarom worden deze 5V signalen omgezet naar een lagere spanning met behulp van de BAV99 diode.



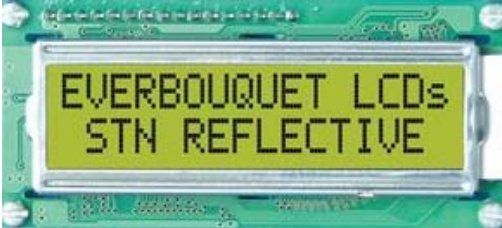
Figuur 19 PS/2 interface

De BAV99 diode is een dubbele kleine signaal diode. De input van de FPGA kan maximum 3,75V kan verdragen. De bovenste diode van de BAV99 zal

geleiden wanneer de spanning op de data- of kloklijn hoger is dan  $3V_3 + V_{forward}$ . Op deze manier zal de FPGA langs deze weg het minste weerstand zien en staat er op de ingang ook de spanning  $3V_3 + V_{forward}$ . Hierbij ondervonden we dat dit dan uitkomt op 4V wat nog geen correcte waarde is om aan de ingang van de FPGA aan te bieden. Echter wanneer nu de interne pull down van de FPGA ingangspin actief wordt gezet krijgen we een ingangsimpedantie van de FPGA van ongeveer 33K in plaats van een normale 500K tot 1M. De combinatie 10K (R44 en R45) en 33K zorgt er nu voor dat we op de input een spanning krijgen van ongeveer 3.72V. Bij een negatieve spanning op de PS/2 pin zal de BAV99 diode geleiden naar massa. In het schema zijn ook pull-up weerstanden voorzien. Dit is omdat indien de datalijn in rust is (geen data over de datalijn), de data- en de kloklijn hoog actief moeten zijn.

## 14.8 LCD

### 14.8.1 Interface

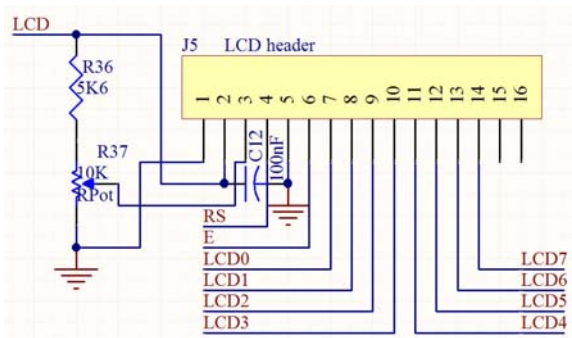


Figuur 20 LCD karakter display

Op het development board is ook een LCD karakter display aansluiting voorzien. Dit is een gewone 16 pins SIL (single in line) female header waar het display in geplugd kan worden. De meeste LCD karakter displays worden naar de buitenwereld geconnecteerd door middel van een Controller IC. Meestal is dit de HD44780 van Hitachi maar het kan ook een equivalente controller zijn. Op de markt zijn er 1x16 (1 lijn \*16 karakters), 2x16, 2x20, 4x20 en 2x40 LCD displays verkrijgbaar. Een LCD kan gebruikt worden om bijvoorbeeld een meetwaarde of tekst weer te geven.

Het LCD is gevoed op 5V. Dit wordt gedaan omdat de meeste types in omloop 5V voeding vereisen. Echter de signalen die naar het LCD gestuurd worden vanuit de FPGA zijn wel 3V3. Dit is voldoende om door het LCD als een logische '1' geïnterpreteerd te worden.

Er zou een probleem ontstaan indien de data van het LCD naar de FPGA toe gelezen zou worden. Dit zouden dan 5V signalen zijn omdat het LCD op 5V gevoed is. Deze 5V signalen zijn niet toegestaan op de ingang van de FPGA. Echter door de Read/Write lijn van de LCD op '0' = ground te hangen zorgen we ervoor dat er enkel data geschreven kan worden.



Figuur 21 LCD interface

Tabel 2 LCD aansluiting

1 <u>V<sub>SS</sub></u>	Massa	5 R/W	0 = schrijven naar LCD 1 = lezen van LCD
2 <u>V<sub>CC</sub></u>	Voeding (+5V)	6 E	1 -> 0 <u>Enable</u>
3 <u>V<sub>ee</sub></u>	Contrastpin	7 - 14	Data bus lijn (7 = LSB)
4 RS	0 = <u>instructie</u> 1 = data	15 - 16	Background <u>light</u> (niet bij alle LCD's)

Pin 1 en 2 verzorgen dus de voeding. Pin 3 is via een potentiometerschakeling aangesloten zodat het contrast van de LCD geregeld kan worden. Met RS wordt bepaald of een instructie of data geschreven wordt. De read/write lijn hangt op massa zodat er dus altijd data of instructies geschreven wordt en dus nooit gelezen. De enable reageert op een negatieve flank. Deze geeft aan dat de datalijnen klaar zijn om data te ontvangen. Er zijn 8 data bus lijnen voorzien waarbij pin 7 de LSB (Least significant bit) is en pin 14 de MSB (most significant bit). Pin 15 en 16 kunnen bij sommige LCD's dienen om een achtergrondverlichting van spanning te voorzien.

### 14.8.2 Protocol

Het protocol dat hier volgt heb ik gebaseerd op de HD44780 controller van Hitachi.

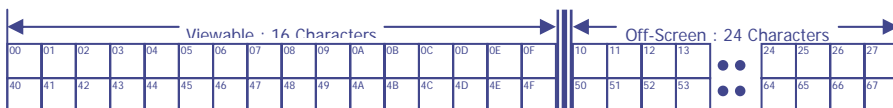
### 14.8.3 Geheugens en registers

Deze controller heeft verschillende interne geheugens. Er worden volgende typen geheugen onderscheiden: RAM-, het CG RAM- en het CG ROM geheugen.

Het display data RAM geheugen (DD RAM) slaat display data op in 8-bit karakter code. De capaciteit van dit geheugen is 80x8 bits wat overeenkomt met 80 karakters. De ruimte die in dit geheugen niet gebruikt wordt kan gebruikt worden als algemeen data RAM geheugen. Het DD RAM is eigenlijk het geheugen dat de data opslaat die op het LCD te zien is. Het verzorgt bij een 2x16LCD (veel gebruikt) 2 lijnen van 16 karakters maar ook 24 "off-screen" karakters.

Elke karakterpositie heeft een hexadecimaal adres. Lijn 1 begint bij adres 00h en eindigt bij 27h waarbij tot en met 0Fh zichtbaar is. Lijn 2 begint bij 40h en eindigt bij 67h waarbij tot 4Fh zichtbaar is.

De "off-screen" karakters kan men gebruiken om andere data in te zetten die later zichtbaar gemaakt kan worden door het display te shiften met een display shift left of een display shift right commando.



Het karakter generator ROM geheugen (CG ROM) genereert 5x8 dot of 5x10 dot karakters van 8-bit karakter codes. Het kan 208 5x8dot karakter patronen en 32 5x10 dot patronen genereren.

CG ROM	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
xxxx0001 (2)	!	1	A	Q	a	q										
xxxx0010 (3)	"	2	B	R	b	r										
xxxx0011 (4)	#	3	C	S	c	s										
xxxx0100 (5)	\$	4	D	T	d	t										
xxxx0101 (6)	%	5	E	U	e	u										
xxxx0110 (7)	&	6	F	V	f	v										
xxxx0111 (8)	'	7	G	W	g	w										
xxxx1000 (1)	<	8	H	X	h	x										
xxxx1001 (2)	>	9	I	Y	i	y										
xxxx1010 (3)	*	:	J	Z	j	z										
xxxx1011 (4)	+	:	K	C	k	c										
xxxx1100 (5)	,	<	L	#	l	l										
xxxx1101 (6)	-	=	M	J	m	j										
xxxx1110 (7)	.	>	N	^	n	^										
xxxx1111 (8)	/	?	O	_	o	_										

Figuur 22 karakter generator ROM geheugen

Het karakter generator RAM geheugen (CG RAM) ten slotte. Hier kan de gebruiker karakter patronen in programmeren. Dit kunnen acht 5x8 dots patronen of vier 5x10 dots patronen zijn.

De controller beschikt over twee 8-bit registers. Namelijk een instructieregister IR en een dataregister DR. Het instructieregister slaat instructiecodes op zoals: display clear, cursor shift, adres informatie voor het DD RAM (display data ram) en CG RAM (character generator RAM). Het dataregister slaat tijdelijk data op om te schrijven in het DD RAM of het CG RAM en om deze data uit te lezen in geval dat deze functie van de LCD gebruikt wordt. Dit laatste is dus niet het geval in onze toepassing.

De busy flag (BF) is een bit die "1" is wanneer het display bezig is met het uitvoeren van een interne instructie. Zo kan men weten of er een nieuwe instructie of data geschreven mag worden. Deze BF wordt op data lijn 7 uitgelezen wanneer RS= "0" en W/R = "1". Aangezien in dit project de R/W op ground hangt kan hier geen gebruik van gemaakt worden en zal men dus de maximum uitvoeringstijd moeten respecteren.

De adres counter (AC) kent adressen toe aan het DD RAM en het CG RAM geheugen. Wanneer een adres instructie (bv: "Set DDRAM address =05H") in het instructieregister wordt geschreven, zal deze de adres counter aanspreken en correct zetten. De adrescounter wordt na het schrijven in DDRAM of CGRAM automatisch met 1 verhoogd.

## 14.8.4 commando's

Instruction	Code									
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Clear display	0	0	0	0	0	0	0	0	0	1
Return home	0	0	0	0	0	0	0	0	1	—
Entry mode set	0	0	0	0	0	0	0	1	I/D	S
Display on/off control	0	0	0	0	0	0	1	D	C	B
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—
Function set	0	0	0	0	1	DL	N	F	—	—
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG
Set DDRAM address	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD
Read busy flag & address	0	1	BF	AC	AC	AC	AC	AC	AC	AC
Write data to CG or DDRAM	1	0	Write data							
Read data from CG or DDRAM	1	1	Read data							

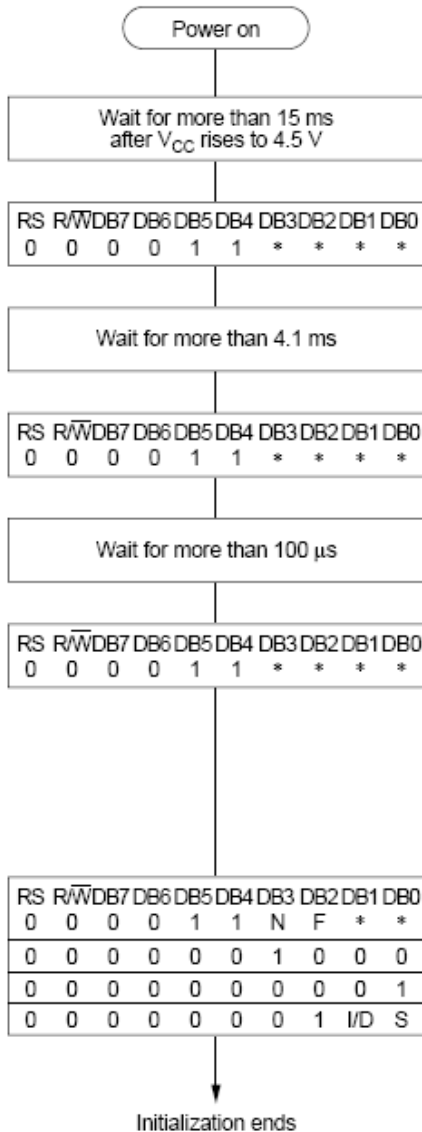
I/D = 1: Increment  
 I/D = 0: Decrement  
 S = 1: Accompanies display shift  
 S/C = 1: Display shift  
 S/C = 0: Cursor move  
 R/L = 1: Shift to the right  
 R/L = 0: Shift to the left  
 DL = 1: 8 bits, DL = 0: 4 bits  
 N = 1: 2 lines, N = 0: 1 line  
 F = 1: 5 × 10 dots, F = 0: 5 × 8 dots  
 BF = 1: Internally operating  
 BF = 0: Instructions acceptable

Figuur 23 Commando-set

Clear Display	Schrijft een blanco patron in alle DDRAM adressen. Zet DDRAM adres 0 in de adres teller. Zet display in originele status als het geshift was.
Return Home	Zet DDRAM adres 0 in adres teller. Zet display in originele status als het geshift was. De DDRAM inhoud wordt niet gewijzigd.
Entry Mode	<p>Incrementeert of decrementeert het DDRAM adres met 1 wanneer er een karakter code in het DDRAM adres gelezen of geschreven wordt. De cursor beweegt naar links bij decrementeren of naar rechts bij incrementeren.</p> <p>Bepaalt de richting van het schiften van het display.</p>
Display On/Off control	Zet het display aan en bepaalt de stijl van de cursor.
Cursor or display shift	<p>Gebruikt om de cursor naar links of rechts te verplaatsen terwijl de adres teller mee aangepast wordt.</p> <p>Het display shiften naar links of rechts. De cursor wordt mee verplaatst en de adres teller blijft hetzelfde.</p>
Function Set	<p>Bepaalt de "... mode" : 4- of 8-bit mode.</p> <p>Bepaalt het aantal lijnen</p> <p>Selecteert het karakter font.</p>
Set CGRAM Adress	Bepaald het CGRAM adres.
Set DDRAM Adress	Bepaald het DDRAM adres. In 1 lijn mode kan dit binnen een adresbereid van 0x00 tot 0x4F en in 2 lijn mode op de eerste lijn van 0x00 tot 0x27 en op de tweede lijn van 0x40 tot 0x67.

### 14.8.5 De initialisatie van het LCD

Wanneer de voeding aan het LCD wordt aangesloten reset het LCD zichzelf. Het zet zich dan in een toestand waarin geen data zichtbaar kan gemaakt worden. Daarom moet het display eerst geïnitieerd worden. Deze initialisatie bepaalt dan hoe het LCD gaan gebruikt zal worden.



Na het aanleggen van een voedingspanning moet er 15 ms gewacht worden tot de spanning gestegen is tot 4,5V.

Hierna moet er 3 maal het function set commando gestuurd worden zodat de display interface op 4 of 8 bit ingesteld worden. De keuze tussen een 1 of 2-lijns LCD, en het dot formaat. De wachttijden zijn hier 4,1 ms en 100 µs.

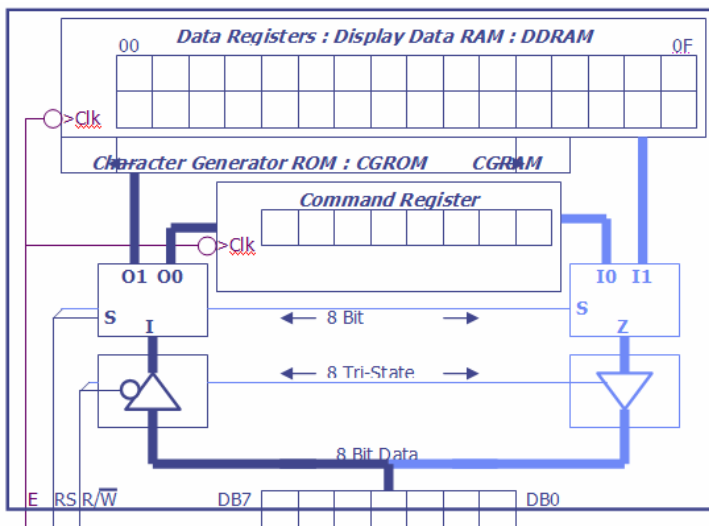
Daarna moet het Display on/off commando bepaald worden. Met wachttijd 37µs na deze instructie. Vervolgens moet het clear display commando verstuurd worden. Hierna moet men weer 37 µs wachten. De laatste te verzenden instructie is de character entry mode instructie met wachttijd 37 µs.



### 14.8.6 De werking

De werking leg ik uit aan de hand van een voorbeeld en afbeelding 5.22 . Willen we bijvoorbeeld een karakter A op de 1<sup>ste</sup> positie links boven schrijven dan moeten we het volgende doen:

1. Het LCD initialiseren
2. De R/W lijn moet nu laag gemaakt worden omdat er geschreven zal worden. Dit enabled de ingangsbuffer en transfereert de data naar de linkse demultiplexer. (Dit is bij ons altijd het geval)
3. De RS lijn moet nu hoog gemaakt worden omdat er data geschreven zal worden en geen instructie. Dit zorgt ervoor dat de demultiplexer de data doorlaat naar het CGROM.
4. De code "01000001" wordt nu op de databus DB0 – DB7 gezet. Uit figuur 5.19 kan afgeleid worden dat het hier dan gaat over karakter "A" gaat.
5. Het CGROM ontvangt het pixel patroon voor karakter "A". Wanneer er nu een puls op de enable ingang gegeven wordt, dan zal bij de flank van hoog naar laag, het pixel patroon doorgeklokt worden naar het DDRAM en zal "A" zichtbaar worden op het LCD.



Figuur 24 Intern logisch schema

Omdat de adres teller nu automatisch verhoogt wordt, is het mogelijk een tweede karakter te sturen. Dit moet natuurlijk wel gebeuren na het respecteren van de verwerkingstijd of door te controleren op de "busy-flag" maar dit is in ons ontwerp niet mogelijk.

## 14.9 RS-232

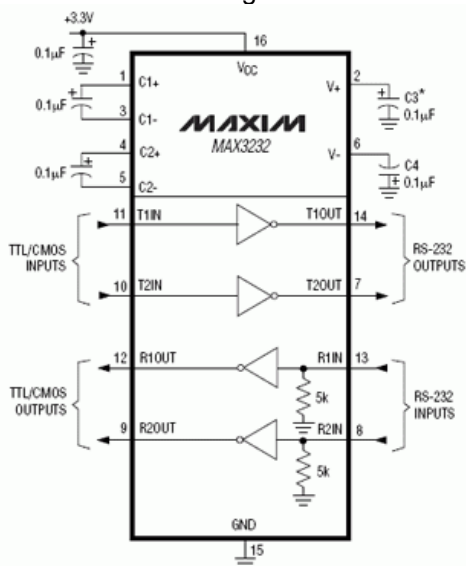
### 14.9.1 Inleiding

RS-232 is in de telecommunicatie een standaard voor seriële binaire data communicatie. Origineel is de standaard ontworpen voor het aansluiten van apparatuur als computers, printers, terminals en dergelijke (DTE: Data Terminal Equipment) op modems (DCE: Data Circuit-terminating Equipment). De tegenwoordige officiële naam van de standaard is ANSI/EIA/TIA-232-F.

Het is een asynchrone communicatie. Dit wil zeggen dat de informatie niet op vooraf afgesproken tijdstippen verzonden wordt. Er kan dus op een willekeurig tijdstip een dataoverdracht gebeuren en het is de taak van de ontvanger om dit te detecteren. De seriële poort wordt ook wel COM (communicatie)–poort genoemd. Ze is bidirectioneel. Dit wil zeggen dat men tegelijk data kan versturen en ontvangen. Volgens de originele standaard is de snelheid gelimiteerd tot 20 kbps, maar in de praktijk blijken veel hogere bandbreedtes mogelijk. Vandaag de dag worden deze poorten meestal weggelaten op computers en vervangen door USB naar RS-232 converters.

### 14.9.2 Interface

De RS-232 interface op het bordje is opgebouwd rond het MAX3232 IC van Texas Instruments uitgevoerd in een SSOP package. Dit is een RS-232 driver/receiver met twee drivers en twee receivers. Er kunnen dus twee RS-232 interfaces aangesloten worden op dit IC. Op het moederbordje wordt er echter maar één van gebruikt. Dit IC kan data rates aan tot 250kbit/s.



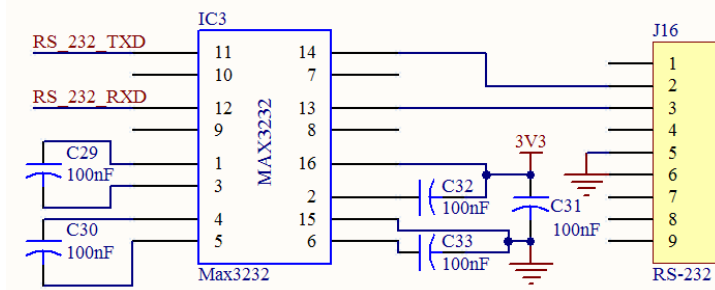
Figuur 25 MAX3232 schema

De functie van dit IC bij het uitsturen van data vanuit de FPGA is een logi-

sche '0' en '1' omzetten naar ongeveer 5,4V ('0') en -5,4V ('1') op de uitgang.

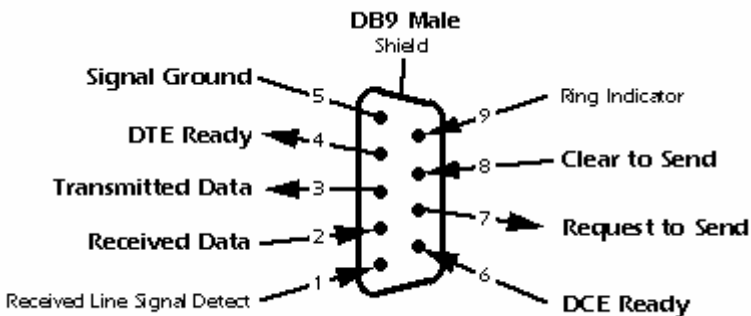
De functie van dit IC bij het lezen van data naar de FPGA is de -3 tot -25V omzetten naar een logische '1' en de 3 tot 25V omzetten naar een logische '0'

Omdat het IC gevoed wordt met 3,3V zal een logische '1' dus ook 3,3V zijn. Verder vraagt dit IC enkel wat ontkoppelcondensatoren.



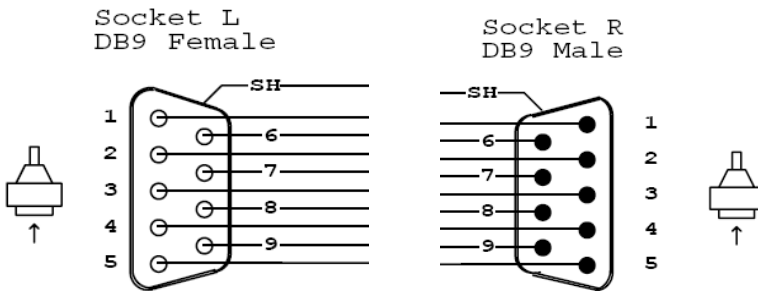
Figuur 26 RS-232 interface

De RS-232 connector is een standaard DB-9 connector van het type female. Figuur 5.25 toont de pinout van een DB9 female connector met bijhorende signalen. Voor een eenvoudige communicatie zijn er echter maar 2 signalen nodig. Namelijk de Transmitted data (TxD) en de Received data (RxD). De andere signalen worden handshake-signalen genoemd en dienen om de status van PC of modem door te geven.



Figuur 27 DB-9 pinout

Voor toepassingen tussen PC en development board wordt een standaard 9-pin DTE-to-DCE seriële kabel gebruikt worden. Dit is een kabel met male- en female-DB-9 connector. Hierbij zijn alle pinnen direct met elkaar verbonden.



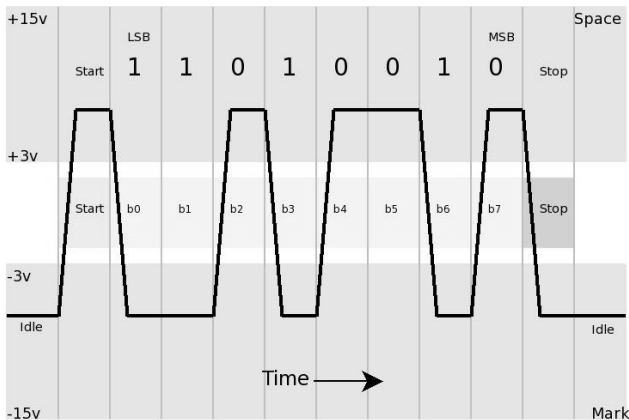
Figuur 28 Standaard DTE naar DCE seriële kabel

### 14.9.3 Protocol

Bij RS-232 communicatie wordt de informatie bit voor bit verstuurd over het fysieke kanaal. Daarom wordt de informatie opgedeeld in datawoorden. De lengte van deze datawoorden kan ingesteld worden tussen 5 en 8 bits maar meestal wordt voor 8 bits gekozen. Het belangrijkste is hier dat dit bij zender en ontvanger gelijk is ingesteld.

De “baudrate” is het aantal bits dat per seconde doorgestuurd wordt. Dit moet men zowel bij zender als ontvanger hetzelfde programmeren. Veel gebruikt is hier 19200 bps.

De lijnspanning heeft 2 toestanden. De aan-toestand (Mark) met spanningen van -25V tot -3V die staan voor een logische 1 en de uit-toestand (Space) met spanningen van 25V tot 3V die staan voor een logische 0. De lijn wordt in de aan-toestand gehouden wanneer er geen data verzonden worden. Afhankelijk van de gebruikte voeding worden veelal signaalniveaus van  $\pm 5$  V,  $\pm 10$  V,  $\pm 12$  V en  $\pm 15$  V gebruikt.



Figuur 29 Signalen RS-232

De data bestaat uit:

- Één startbit die altijd van het space-niveau is zodat hier gemakkelijk op gedetecteerd kan worden door de ontvanger omdat de lijn ongebruikt in mask-niveau toestand is. Na het ontvangen van deze eerste bit berekent de ontvanger wanneer de volgende databits kunnen ontvangen worden.
- De databits met de MSB (Most Significant Bit) als eerste.
- Mogelijk een pariteitsbit. Deze wordt gebruikt om fouten te detecteren. De zender berekent de waarde van deze bit aan de hand van de verzonden informatie. De ontvanger voert dezelfde berekening uit en controleert dan of het ontvangen pariteitsbit overeenkomt met de berekende waarde. Is dit niet het geval dan is er een transmissiefout opgetreden en worden de data niet gebruikt.
- Één stopbit met mark niveau. Als de ontvanger een andere waarde dan mark detecteert op het moment dat het stopbit had moeten worden ontvangen, dan is daarmee gedetecteerd dat er een fout heeft plaatsgevonden.

## 14.10 VGA

### 14.10.1 Inleiding

In 1987 ontwikkelde IBM de VGA (Video Graphics Array)-standaard. Dit is een standaard voor het weergeven van beelden op een computermonitor. Deze originele VGA standaard had een resolutie van 640x480 pixels. VGA maakt gebruik van 3 kleuren: rood, groen en blauw (RGB). Deze aansturing gebeurt analoog. Er zijn ook nog 2 synchronisatiesignalen die digitaal TTL-compatibel zijn.



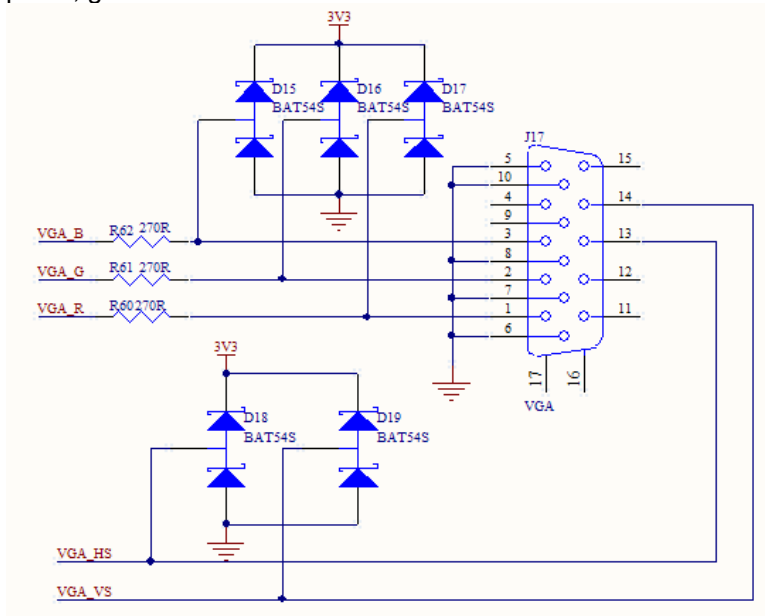
Pin	Functie		
1	Red video	7	Green ground
2	Green video	8	Blue ground
3	Blue Video	10	Sync ground
5	Ground	13	Horiz. Sync
6	Red ground	14	Vert. Sync

Figuur 30 VGA connector

### 14.10.2 Interface

Zoals in de inleiding vermeld zijn er 5 stuursignalen nodig. Een horizontaal en verticaal digitale synchronisatiesignaal. De 3 kleursignalen zijn analoge signalen met een peak-to-peak niveau van 0,7V tot 1,0V. De waarde van dit analoog signaal bepaalt de sterkte van de kleur. Echter om de interface simpel te houden worden deze toch gestuurd met digitale signalen. We kunnen

zo  $2^3 = 8$  kleuren creëren. Deze zijn: zwart, blauw, groen, cyaan, rood, paars, geel en wit.



Figuur 31 VGA interface

Deze digitale signalen worden omgevormd naar ongeveer 0.7V. Met gebruik van 3V3 I/O's wordt dit gedaan door 3 weerstanden van 270Ω. Deze staan dan in serie met de 75Ω weerstanden van de monitor ingangen. Zo ontstaat een spanningsdeler met een uitgangsspanning van  $3.3 * 75 / (270 + 75) = 0.72V$ .

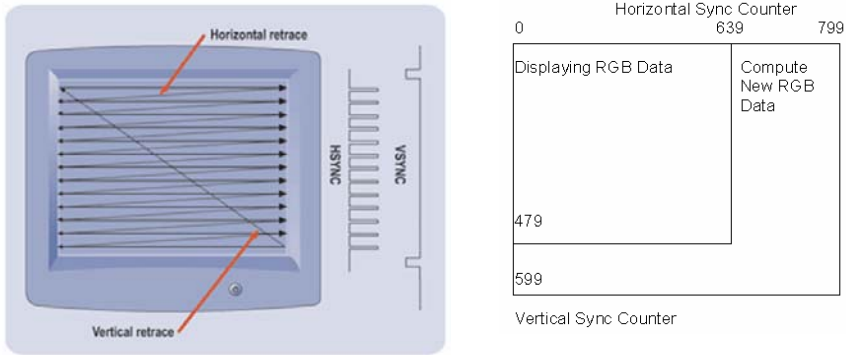
Wanneer een VGA kabel op de connector aangesloten wordt, kan er een hoge ingangsspanning op de FPGA-pinnen ontstaan ten gevolge van inductiespanningen. Om deze te hoge spanning op de ingangspin te voorkomen wordt gebruik gemaakt van BAT54S diodes. Dit zijn dubbele schottky barrier diodes met een spanningsval van ongeveer 300mV tot 400mV.

De bovenste schottky zal geleiden wanneer de spanning op de datalijn hoger is dan  $3V3 + V_f$  zodat de FPGA ingangspin ook deze spanning ondervindt. De onderste schottky wordt gebruikt voor hetzelfde principe maar dan bij negatieve spanningen zodat de ingangspin  $-V_f$  ondervindt.

### 14.10.3 Protocol

Een RGB elektronenbundel schrijft het beeld op het scherm bestaande uit een aantal horizontale lijnen. Iedere lijn wordt gevormd door 640 pixels en één beeld bestaat uit 480 lijnen (640x480 pixels). Per seconde worden 60 beelden op de monitor gezet. Dit leidt tot een frequentie van 60 Hz die hoger

is dan de menselijke detectie van flikkering (30 Hz refreshrate). Het signaal HSYNC geeft aan wanneer er een volledige lijn is geschreven. VSYNC geeft aan dat er een volledig beeld geschreven is.



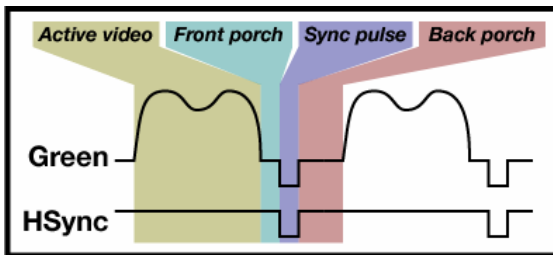
Figuur 32 Het HSYNC- en het VSYNC TTL-sigitaal

**Een lijn:**

Omdat een RGB-elektronenbundel een zekere tijd nodig heeft om terug te keren (row en guard bands) duurt de scanline time 31,77µs.

Dit komt overeen met  $31.77 \text{ us} * 25175000 = 800 \text{ pixels/line}$ .

Elke lijn begint met een stuk actieve video. Hier worden de RGB waardes voor elk pixels gestuurd. Hierna komt er een blanco stuk waar er zwarte pixels verzonden worden. In het midden van dit interval wordt er een horizontale synchronisatiepuls verzonden. Het lege interval voor de sync puls wordt het front porch genoemd. Het lege interval na de sync puls wordt de back porch genoemd. De porch gedeelten waren origineel ontwikkeld om de trage elektronica in de oude televisies tijd te geven om zich voor te bereiden op de actieve video. Na de actieve video wordt het elektronenkanon terug gekeerd naar het begin van een lijn. Hiervoor moet er voor en na de actieve video ook nog een bepaalde tijd gewacht worden. Dit wordt de top- en bottom border genoemd (niet zichtbaar in tekening).



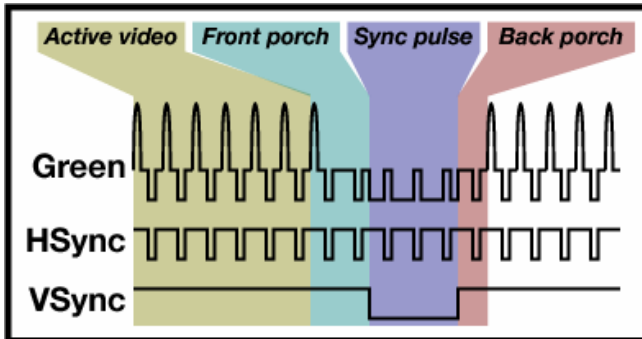
Figuur 33 Een overzicht van de opeenvolgende signalen voor de opbouw van 1 lijn

Tabel 3 Timing van een lijn

Signaal	Tijdsduur	Overeenkomstige pixels
Left border	0.94µs	8
Active video	25.17µs	640
Right border	0.94µs	8
Front porch	0.94µs	8
Sync pulse	3.77µs	96
Back porch	1.89µs	40

**Een beeld:**

Dit is hetzelfde principe als bij het schrijven van een lijn. Echter worden er andere tijden gebruikt en hebben we een top- en bottom border.



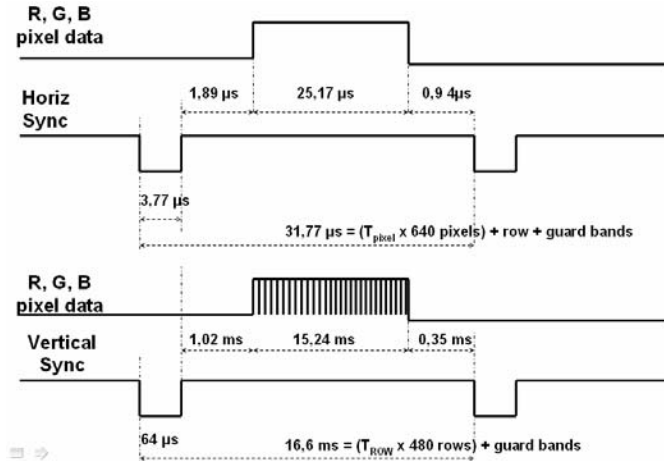
Figuur 34 Een overzicht van de opeenvolgende signalen voor de opbouw van 1 beeld

Tabel 5.4 Timing van een beeld

Signaal	Tijdsduur	Overeenkomstige pixels
Top border	1,4 ms	8
Active video	15,25 ms	480
Bottom border	1,4 ms	8
Front porch	0,35 ms	2
Sync pulse	0,06 ms	2
Back porch	1,02 ms	25



Dit alles komt overeen met een totaal van 525 lijnen per beeld. Dit leidt tot een frequentie van  $800\text{pixels} \times 525\text{pixels} \times 60\text{hz} = 25,2\text{ MHz}$ . Elke pixel neemt dus ongeveer  $40\text{ns}$  in beslag.



Figuur 35 Tijden van de horizontale- en de verticale timing

## 14.11 Infrarood

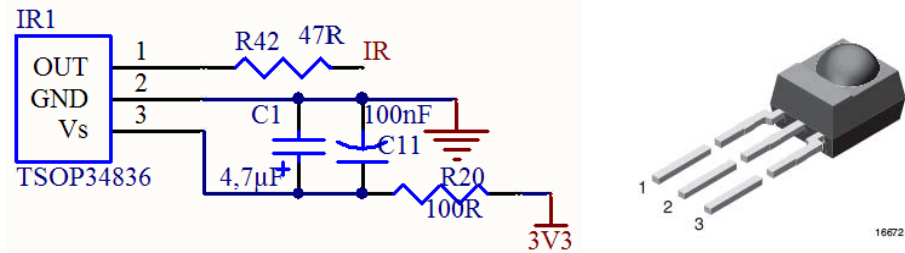
### 14.11.1 Inleiding

Infrarood (IR)- licht wordt in de elektronica wereld vooral gebruikt om gegevensoverdracht via afstandsbedieningen te voorzien. Infrarood licht heeft een frequentie die lager is dan de gevoeligheid van het menselijk oog zodat we dit niet met het blote oog kunnen zien. Het geeft echter wel een bepaalde warmte af die voelbaar is. Omdat andere lichtbronnen, bijvoorbeeld de zon en lampen, ook infrarood licht kunnen uitstralen kan dit de communicatie verstoren. Om deze interferentie te vermijden wordt bij afstandsbedieningen het infrarood licht gepulseerd en de ontvanger afgestemd op een bepaalde frequentie zodat alle andere frequenties genegeerd worden. Deze frequentie ligt tussen de 30 en 60kHz waarbij 36kHz de meest gebruikte is. Voor de zender wordt een IR led gebruikt en voor de ontvanger een IR diode.

Door gebrek aan een standaard hebben de meeste fabrikanten hun eigen protocol ontwikkeld. Maar het meest gebruikte protocol is de RC-5 codering van Philips. Dit is later uitgebreid naar RC-6.

### 14.11.2 Interface

De infrarood interface is opgebouwd rond de TSOP34836 IR ontvanger van Vishay. Het 36kHz gedecodeerde signaal kan direct verwerkt worden door een processor. Het is de bedoeling hiermee het signaal afkomstig van een televisie-afstandsbediening in te lezen in de FPGA.

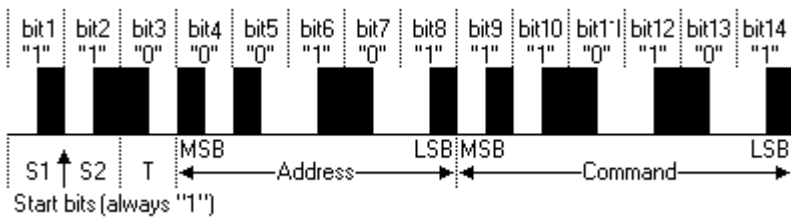


Figuur 36 IR interface

R20 en C1 moeten zoals in de datasheet voorzien worden om voedingstoringen te voorkomen. C11 is nog extra voorzien als ontkoppelcondensator. Weerstand R42 is een serie-terminatieweerstand van 47Ω. Dit heeft tot doel het reduceren van ringing/overshoot/undershoot op de lijn ten gevolge van niet aangepaste impedanties. Deze techniek wordt in de praktijk toegepast voor HF (Hoog frequent) – toepassingen of zeer kritische signalen. Hierbij wordt de weerstand dan exact uitgerekend. In ons geval is de serieweerstand een voorbeeld van degelijk ontwerpen. De norm is hier een weerstand van 50 a 150Ω zodat de output van de driver voldoende gedempt wordt om reflecties te vermijden.

### 14.11.3 Protocol RC-5

Het RC-5 protocol maakt gebruik van bifase codering (ook Manchester codering genoemd) met een IR draaggolf van 36kHz. In manchester encoding is de lengte van 1 bit opgedeeld in 2 helften. De ene helft heeft een bepaald spanningsniveau terwijl de andere helft een ander spanningsniveau heeft. Door de wisseling in het midden van een bit is synchronisatie eenvoudig. Een "1" wordt bij RC-5 voorgesteld door een stijgende flank in het midden van een bit terwijl een "0" met een dalende flank wordt voorgesteld. Één bit heeft een lengte van 1.778ms. De helft van deze bit is dus gevuld met pulsen van de 36kHz draaggolf. Een boodschap bestaat uit 14 bits wat overeenkomt met een totale lengte van 24.892ms.



Figuur 37 RC-5 protocol

De eerste 2 bits zijn start bits die altijd "1" zijn. Deze worden gevolgd door een toggle bit. Deze bit wordt geïnverteerd elke keer een knop terug ingedrukt wordt. Dit wordt gedaan om onderscheid te kunnen maken tussen een knop die ingedrukt blijft of een knop die herhaaldelijk wordt ingedrukt. De volgende 5 bits stellen het adres van het IR apparaat dat we willen bedienen. De MSB wordt hier eerst verzonden. De laatste 6 bits stellen het commando voor met de MSB eerst verzonden.

Zolang een toets ingedrukt blijft zal de boodschap elke 114ms opnieuw verzonden worden terwijl de toggle bit hetzelfde blijft.

De decodering kan het beste gebeuren door de waarden in te lezen in het midden van de tweede bithelft omdat men hier de juiste waarde van de bit heeft.

## 14.12 S/Pdif

### 14.12.1 Inleiding

S/Pdif staat voor Sony/Philips Digital Interface Format. Het is een standaard om digitale audio te verzenden. Het is de consumentenversie van de standaard die bekend is als AES/EBU (Audio Engineering Society, European Broadcastint Union). Dit is een standaard die gebruikt wordt bij datatransmissie in onder andere professionele studio-opnamen. S/Pdif heeft een paar kleine veranderingen en wordt gebruikt in goedkopere hardware toepassingen zoals cd-spelers en dvd-spelers.

De bekabeling van S/Pdif kan zowel optisch als elektrisch gebeuren. Optische bekabeling heeft echter het grote voordeel dat het signaal immuun is voor elektrische storingen. Bij elektrische bekabeling wordt gebruik gemaakt van een 75ohm coaxiale kabel met RCA-connector. Bij optische bekabeling wordt gebruik gemaakt van een TOSLINK connector.



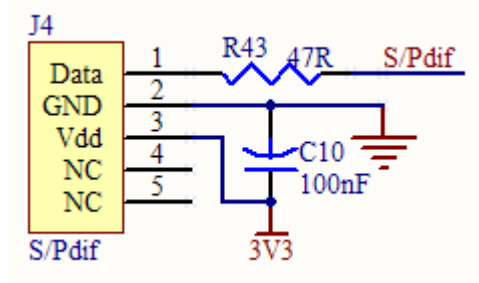
Figuur 38 Toslink connector



RCA connector

### 14.12.2 Interface

De interface is zeer eenvoudig opgebouwd. De TORX147PL connector voor TOSLINK connectie wordt gevoed op 3V3 en voorzien van een ontkoppelcondensator. Ook hier is er een serie-terminatieweerstand voorzien.

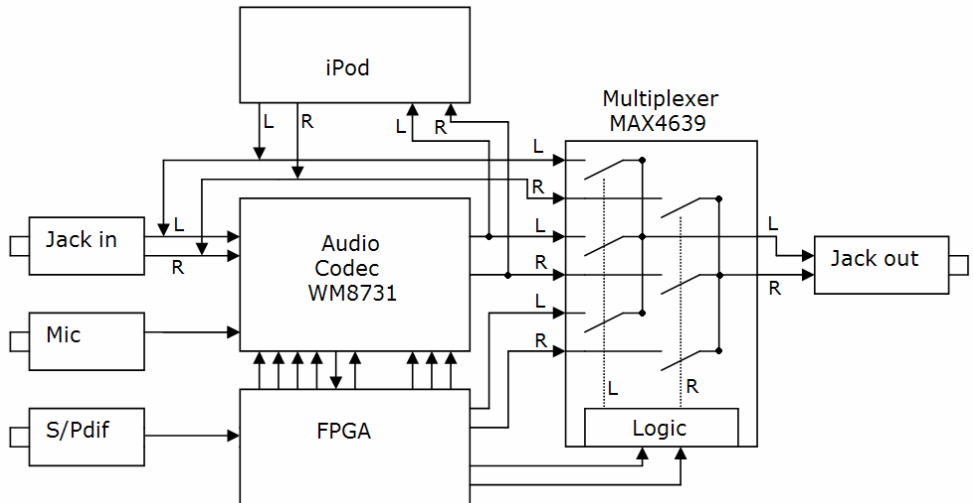


Figuur 39 S/Pdif interface

## 14.13 Audio gedeelte

### 14.13.1 Opbouw

Op het development board is gekozen om een vrij uitgebreide audio gedeelte. Het audio gedeelte is opgebouwd rond de WM8731 Audio Codec van Wolfson die bestuurd wordt door de FPGA.

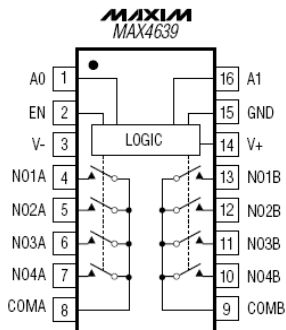


Figuur 40 Functioneel overzicht audio gedeelte

De ingangen van het audio gedeelte zijn de volgende:

- S/Pdif: Deze digitale audio bitstream gaat rechtstreeks naar de FPGA.
- Mic: Dit is een ingang voor een microfoon. Het monosignaal hangt aan een speciaal daarvoor voorziene ingang van de codec.
- Jack in & iPod line out: Deze ingangen hangen samen. Dit zijn stereo ingangen voor de codec. Hier wordt de mogelijkheid geboden om een keuze te maken tussen de iPod als ingangssignaal of om een externe mp3-speler, MD, compact-disc speler, ... te koppelen als ingangssignaal aan de "jack in".

Er is gekozen om via een analoge multiplexer, de MAX4639 van Maxim, om een keuze te maken tussen de verschillende uitgangen. En deze keuze naar één audio output jack te sturen.

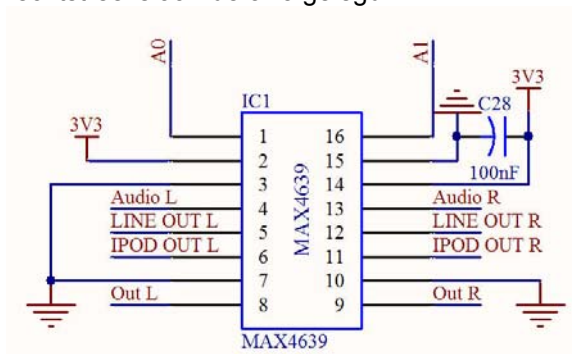


MAX4639 (Dual 4-to-1 Mux)

A1	A0	EN	COMA	COMB
X	X	0	None	None
0	0	1	NO1A	NO1B
0	1	1	NO2A	NO2B
1	0	1	NO3A	NO3B
1	1	1	NO4A	NO4B

Figuur 41 Analoge multiplexer MAX4639

Deze keuze wordt gemaakt door A0 en A1 van de multiplexer met logische signalen te bedienen door middel van twee datalijnen van de FPGA. De "A" ingangen van de multiplexer gebruiken we voor de linkse kanalen van de stereo audio en de "B" ingangen voor de rechtse kanalen. De enable is rechtstreeks aan de 3V3 gelegd.



Figuur 42 Schema MAX4639

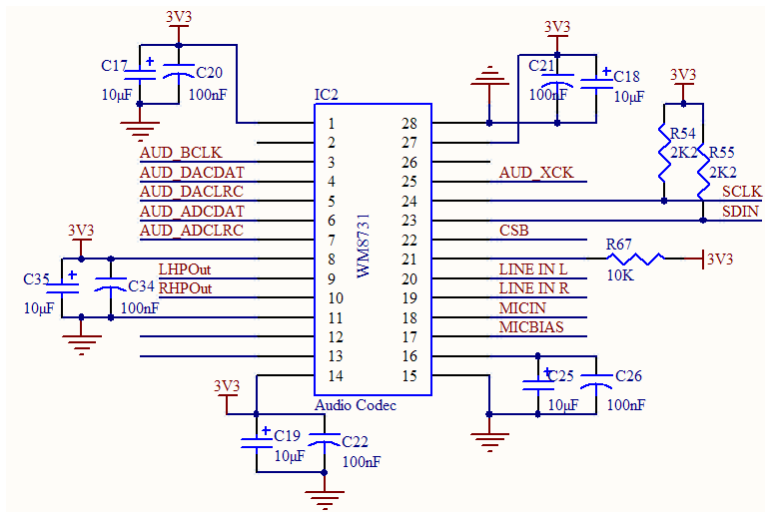
Er kan gekozen worden tussen de volgende signalen naar de uitgang:

- L en R (links en rechts) outputs van de FPGA: Dit kan gebruikt worden om audio digitaal in PWM verwerkt uit te sturen. Hiervoor moet A0 = "0" en A1 = "0" zijn.
- L en R line outputs van de audio codec. Hiervoor moet A0 = "1" en A1 = "0" zijn.
- L en R line outputs van de iPod of ander device aan input jack: Zo kan dit audio signaal ook op de uitgang verkregen worden zonder hierdoor de codec te configureren. Hiervoor moet we A0 = "0" en A1 = "1" zijn.

De outputs van de codec zijn ook met de line inputs van de iPod verbonden. Zo kan de iPod dit signaal opnemen. Dit is echter niet bij elke generatie iPod het geval.

Wat zijn nu de mogelijkheden van dit systeem?

1. Het iPod geluid (of dat van een ander toestel op de audio input jack aangesloten) naar de output sturen. Hiervoor moet enkel de multiplexer geconfigureerd worden.
2. Deze audio bewerken en naar de output sturen. De codec digitaliseert dan het signaal, stuurt het naar de FPGA. De FPGA verwerkt dit signaal terug naar de codec die het digitaal naar analogo omzet en op zijn uitgang zet. De multiplexer selecteert nu het codec output signaal.
3. De microfooningang al of niet bewerkt naar de output sturen. Op dezelfde manier als in punt 2 omschreven.
4. S/Pdif audio op de output verkrijgen. Dit kan op twee manieren. Enerzijds door het signaal te verwerken door de FPGA en dit naar de codec te sturen, die het op zijn beurt omzet naar analoge audio. Anderzijds door het signaal in PWM op de hiervoor voorziene audio uitgangen van de FPGA te sturen.



Figuur 43 Schema codec

Wat hier volgt is de externe logica voor het aansluiten van de codec. Dit is gebaseerd op de informatie uit de datasheet van de Wolfson WM8731 codec.

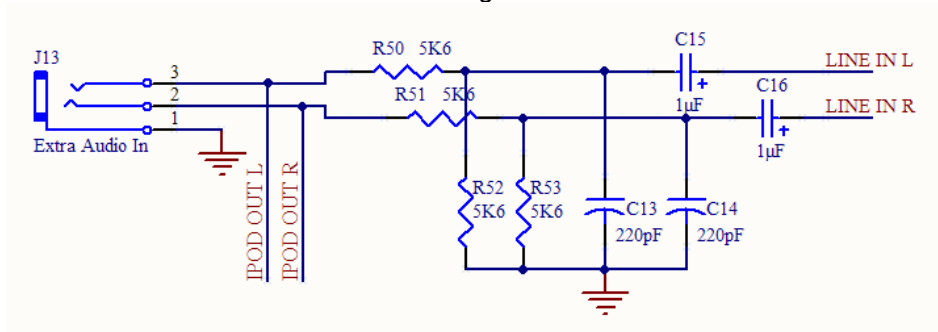
### 14.13.2 Line inputs

J13 in figuur 44 stelt hier de input jack voor. Deze hangt dus zoals gezegd samen met de line-outputs van de ipod. De twee lijn ingangen zijn symmetrisch dus wordt alleen linkse besproken.

De combinatie R50-R52 vormen een spanningsdeler. Deze wordt gebruikt om het ingangssignaal te verminderen naar een 1Vrms-niveau. Dit wordt gedaan omdat de uitgang van bijvoorbeeld een CD-speler een 2Vrms-

niveau heeft. R52 wordt als ontladingspad voor C15 gebruikt. Dit beveiligt de input tegen C2 die zich te hoog oplaadt en geconnecteerde apparatuur kan beschadigen

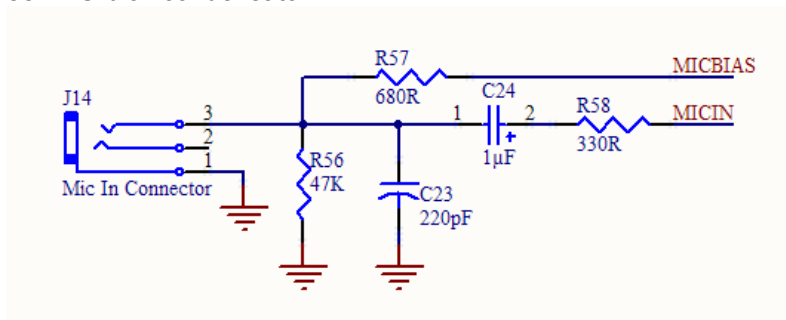
C13 vormt een laagdoorlaat filter voor RF interferentie opgepikt door kabels. C15 vormt een DC-blok condensator om het mogelijke DC-signaal tussen audio apparatuur en de codec te blokkeren. Samen met de ingangsimpedantie van de codec vormt C15 een hoogdoorlaatfilter.



Figuur 44 Line inputs

### 14.13.3 Microfoon ingang

De microfoon ingang kan gebruikt worden om een brede waaier van mono-microfoons te gebruiken. Micbias is noodzakelijk om de mogelijkheid te bieden om electret microfoons (kleine condensatormicrofoons met ingebouwde versterker, die een voedingsspanning nodig hebben) aan te sluiten. R57 en R56 vormen samen een deel van het micbias netwerk. C23 vormt samen met de source impedantie van de microfoon een RF filter. C24 is hier weer een DC-blok condensator.

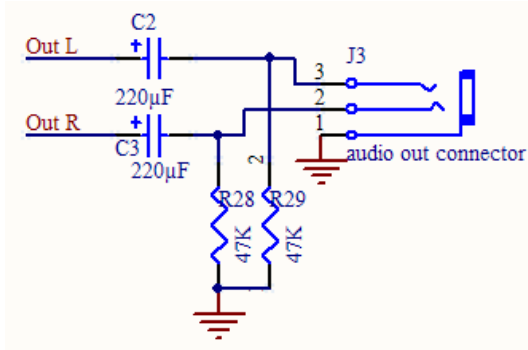


Figuur 45 Microfoon ingang



#### 14.13.4 Output

Dit is dus het gedeelte na de multiplexer. Hier is gebruik gemaakt van de LHPOUT en RHPOUT van de codec. Dit zijn de headphone uitgangen die speciaal ontworpen zijn voor het aansturen van 16 of 32 ohm hoofdtelefoons.



Figuur 46 Uitgangen

C2 vormt hier een DC blok condensator. R28 vormt een pull down weerstand om C1 te ontladen om te voorkomen dat de spanning op de hoofdtelefoon tot een schadelijke spanning stijgt.

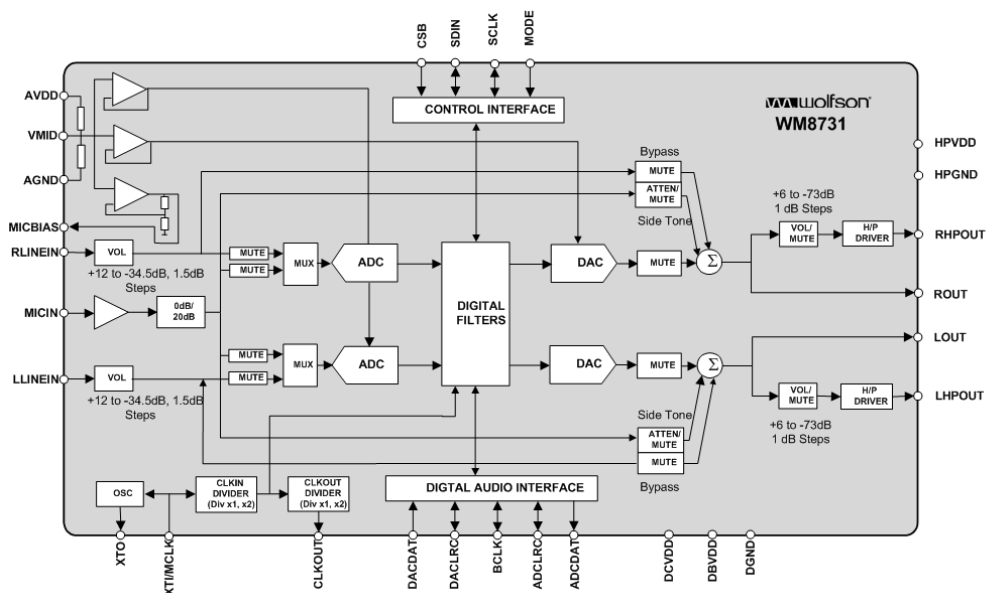
#### 14.13.5 Audio codec

Er is gekozen voor de 24bit-audio codec een WM8731 codec van Wolfson micro-electronics. Dit is een 28pins SSOP package IC. Het is een codec die speciaal ontworpen is voor draagbare audio producten zoals MP3-spelers en MD-spelers.

De codec heeft lijn-ingangen en een microfoon ingang naar de ADC (Analoog to Digital Converter) en hoofdtelefoon uitgang en lijn-uitgangen afkomstig van de DAC (Digital to Analog Converter). De codec beschikt daarnaast over een interne klok generator en een configureerbare audio interface. Het device bevat mute-opties (dempen) en standby mogelijkheden. Dit alles onder software controle. De WM8731 heeft een zeer laag verbruik. De codec kan sample rates van onder andere 44.1kHz, 48kHz, 96kHz, ... verzorgen.

De gedigitaliseerde audio (audio interface) kan gebruik maken van verschillende audio formaten: I<sup>2</sup>S, DSP mode, MSB-first left justified en MSB-first right justified.

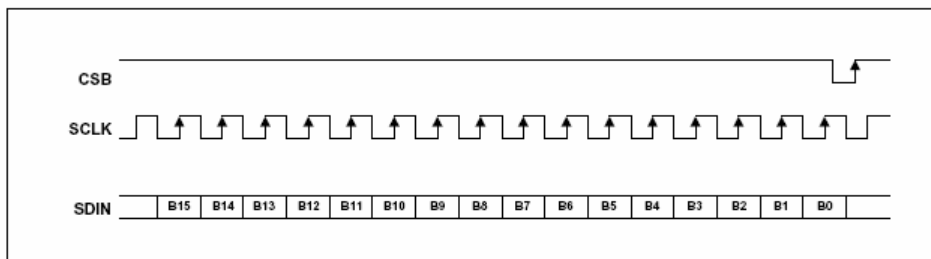
In de hieronder volgende punten worden de belangrijkste elementen van deze codec aangehaald die van toepassing zijn.



Figuur 47 Blokschema WM8731 codec

### 14.13.6 Software controle interface

De codec kan softwarematig geconfigureerd worden in 3-lijns of 2-lijns MPU interface. Er is gekozen voor een 3-lijns interface omdat deze configuratie eenvoudiger leek dan de 2-lijns configuratie. Deze keuze wordt gerealiseerd door de MODE pin van de IC via een weerstand van 10k aan 3.3V te hangen.



Figuur 48 3-wire mode

In deze mode is de SDIN pin bedoeld als programmadata. SCLK wordt gebruikt om de data in te kloppen en CSB om de programmadata in te lachen. Dit lachen wordt gedaan op een stijgende flank van CSB.

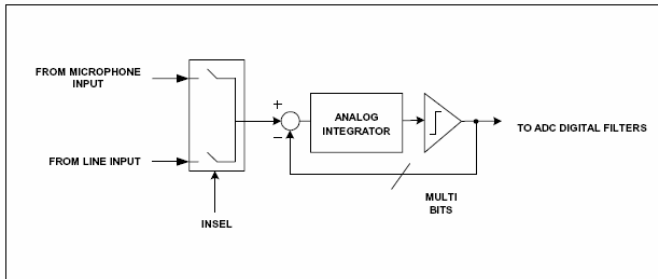
Hierbij zijn:

- B[15:9]: Controle adres bits
- B[8:0]: Controle data bits

Om de codec dus te programmeren wordt er eerst een adres bepaald van het register dat we willen configureren, gevolgd door de data die er in dit register weggeschreven worden.

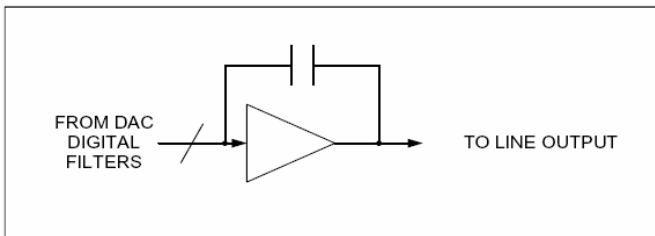
#### 14.13.7 ADC en DAC

De codec maakt gebruik van een multi-bit oversampled sigma-delta ADC. Één van de kanalen is in figuur 49 voorgesteld. Met INSEL wordt de keuze gemaakt tussen de microfoon input of de line input.



Figuur 49 ADC

De DAC is van het type multi-bit sigma delta oversampling DAC. Deze DAC zet de multi-level digitale audio data stream om in hoge kwaliteit analoge audio.

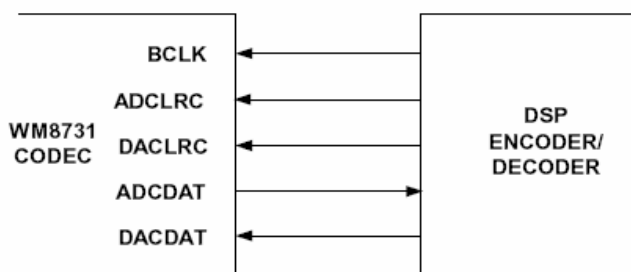


Figuur 50 DAC

### 14.13.8 Audio interface

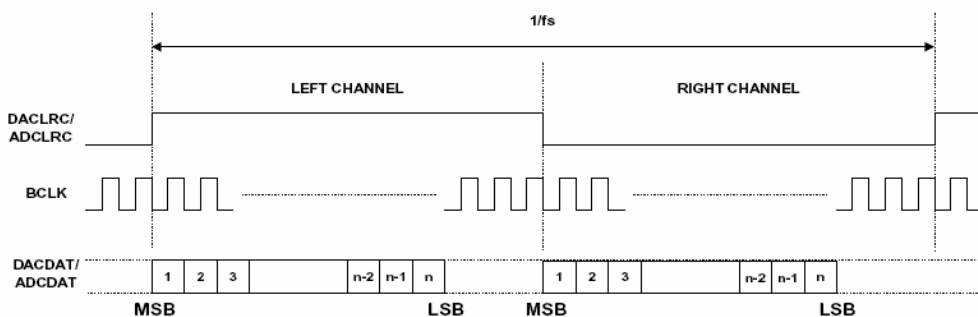
De codec kan gebruikt worden als “master” of “slave”. Master wil zeggen dat de codec een andere DSP (Digital Signal Processor) kan sturen. Slave wil zeggen dat de codec gestuurd wordt door een DSP.

In dit ontwerp is de codec gebruikt als slave met als DSP de FPGA. De codec is default op slave ingesteld.



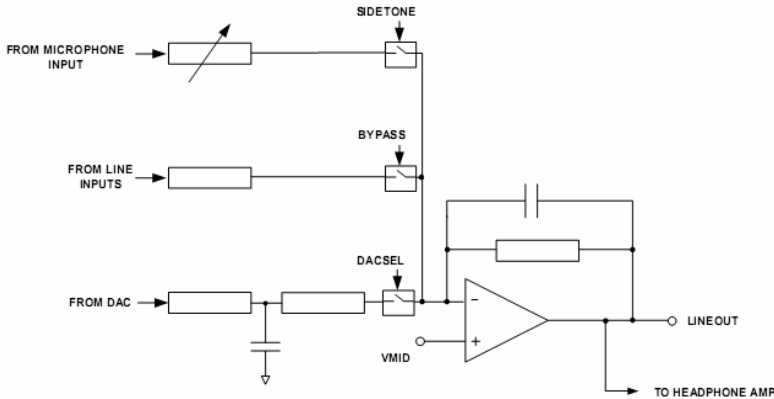
Figuur 51 *codec als slave*

De digitale audio interface neemt de data van de interne ADC digital filter en plaatst deze data op de ADCDAT (Analoog Digitaal Converter Data) output. Dit is dus het geluid dat omgezet is naar een gedigitaliseerd formaat (right justified, left justified, I<sup>2</sup>S, DSP mode). ADCDAT is de data stream output van de ADC filter met links en rechts kanaal aan elkaar gemultiplexeerd. ADCLRC (ADC sample rate Left/Right Clock) is een klok die controleert of de data op de ADCDAT lijn afkomstig is van het linkse of het rechtse kanaal. Als synchrone klok wordt hier BCLK (Bit clock) gebruikt. Elke datatransmissie vindt plaats bij een negatieve flank van de BCLK lijn. (zie figuur 52) DACDAT is een ingang van de codec. Dit is de digitale audio data stream die naar de DAC digitale filter gaat. Dit is ook weer met links en rechts kanaal aan elkaar gemultiplexeerd. DACLRC is hier de klok die het kanaal verduidelijkt. En beiden zijn hier ook synchroon met BCLK.



Figuur 52 Voorbeeld met left justified

### 14.13.9 Modes



Figuur 53 Verschillende modes

Zoals in afbeelding 53 te zien zijn er 3 belangrijke modes voor het uitsturen van audio. Hierdoor worden er verschillende signalen naar de uitgangen van de codec gestuurd.

- Met de mode sidetone wordt het signaal van de microfoon-input onbewerkt doorgestuurd naar de uitgang.
- Met de mode bypass worden de signalen van de line inputs rechtstreeks doorgestuurd naar de uitgangen.
- Met de mode From Dac wordt het door de DAC bewerkte signaal doorgestuurd naar de uitgangen.

### 14.13.10 Registers

Om de codec onder software controle te configureren zijn er verschillende registers beschikbaar met hun bijhorende functies. Het adres van een register bestaat uit 7 bits. Het register zelf is 9 bits breed. De belangrijkste registers en functies die voor dit ontwerp van toepassing kunnen zijn worden hieronder kort besproken.

Tabel 4 Registers

<a href="#">Left- &amp; Right Line in</a>	Dit zijn 2 registers die de <a href="#">line inputs</a> configureren. Het ingangsvolume kan ingesteld worden en het signaal kan <a href="#">ge-mute</a> worden.
<a href="#">Left- &amp; Right headphone out</a>	Dit zijn 2 registers die de <a href="#">headphone outputs</a> configureren. Het uitgangsvolume kan hiermee ingesteld worden.
<a href="#">Analogue audio path control</a>	Hiermee kan men de microfoon ingang <a href="#">muten</a> , selecteren of de <a href="#">line input</a> of de <a href="#">mic</a> ingang naar de ADC gaan, <a href="#">bypass</a> inschakelen, DAC inschakelen en de <a href="#">side tone</a> inschakelen
<a href="#">Digital audio interface format</a>	Bepaling van het audio formaat en de datalengte.
<a href="#">Sampling control</a>	Allerlei settings over de sampling
<a href="#">Active control</a>	Interface actief maken
<a href="#">Reset register</a>	De <a href="#">codec resetten</a> .
<a href="#">Power down control</a>	De verschillende onderdelen uitschakelen om energie te besparen.

## 14.14 iPod interface

### 14.14.1 Inleiding



Figuur 54 Enkele iPod modellen

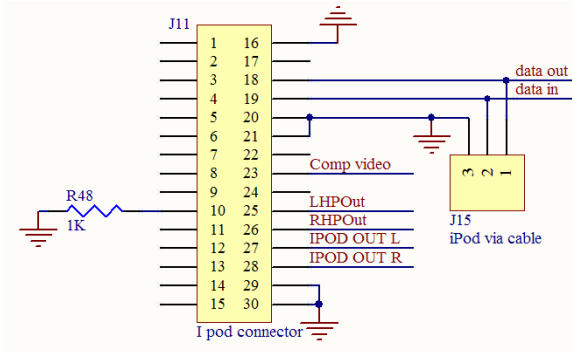
iPod is een merk van draagbare media spelers ontwikkeld door Apple. De eerste iPod werd in 2001 op de markt gebracht. Sindsdien is de iPod een ontzettend populaire digitale audio speler geworden. Er zijn verschillende modellen van op de markt gebracht. De iPod classic, nano, touch, shuffle. De iPod classic slaat zijn gegevens op een harde schijf op terwijl de anderen gebruik maken van flash geheugen. Het overbrengen van muziek naar de iPod wordt verzorgd door de speciale software iTunes.

Sinds de derde generatie iPods werden ze van dock connectors voorzien. Dit zijn 30 pins connectors voorzien in de iPod die het mogelijk maken allerlei gegevens uit te wisselen. Dit bracht extra mogelijkheden om data en geluid met de iPod uit te wisselen. Via deze dock connector kan de iPod ook opgeladen worden. Hierdoor werden er verschillende accessoires op de markt gebracht. Dit gaat van televisies tot speaker-systemen. De bekendste is het iPod docking station. Dit is een apparaat dat is voorzien van een dock connector waarop de iPod wordt gezet, boxen en infrarode afstandsbediening. Zo kan men de muziek beluisteren en bedienen van op afstand.

Op het development board is daarom ook een iPod dock connector voorzien zodat men de FPGA kan laten communiceren met de iPod. Zo kan met behulp van de IR-ontvanger een eigen dock station gemaakt worden.

### 14.14.2 Interface

De dock connector die we gekozen hebben is van het type edge. Dit wil zeggen dat de component aan de zijkant van de print bevestigd wordt met de bovenste rij pinnen op de top layer van de print gesoldeerd en de onderste rij pinnen aan de bottom layer.



Figuur 55 iPod interface

Tabel 5 Pinout dock connector

Pin	Signaal	Functie	Pin	Signaal	Functie
30	GND		23	Video Out	Composite video output
29	GND	Audio en Video massa	20	GND	
28	Right	Audio rechts output	19	Tx	iPod Tx/D lijn
27	Left	Audio links output	18	Rx	iPod Rx/D lijn
26	Right In	Audio rechts input	10	Accessory Indicator/Serial enable	Verbonden met 1K $\Omega$ geeft een piepsignaal bij het inpluggen.
25	Left In	Audio links input			

Data out en data in op het schema zijn bekeken vanuit FPGA kant, data out is dus verbonden met Rx/D van de iPod en data in met Tx/D van de iPod. Er is ook een footprint voor een extra header voorzien zodat men deze data-signalen gemakkelijk met de scoop kan meten.

IPOD OUT L en R zijn verbonden met de extra ingangsjack en gaan zo via de externe componenten naar de codec. Deze bevatten dus de analoge audio afkomstig van de iPod.

Pin 26 en 27 tenslotte zijn verbonden met de uitgang van de audiocodec. Dit zijn namelijk ingangspinnen die toelaten iets op te nemen op de iPod.

Pin 23 geeft een composite video signaal wanneer er een diashow actief is op de iPod. Daarom hebben we ook een RCA video jack voorzien op ons development board.

### 14.14.3 Protocol

Het protocol voor seriële communicatie met de iPod wordt "The Apple Accessory Protocol" genoemd. Het kan gebruikt worden vanaf de 3<sup>de</sup> generatie iPods. Dit protocol is niet vrijgegeven door Apple maar door vele hobbyisten met dezelfde bedoeling als ons is er heel wat te vinden op het internet.

De connectie gebruikt het standaard 8N1 protocol met een baudrate van 19200. Elk pakket bestaat uit 10bits. Namelijk een lage startbit, 8 databits en een hoge stop bit. De data worden verstuurd met LSB eerst. De datalijn staat hoog wanneer deze in rust is.

Een boodschap naar de iPod en verzonden door de iPod bestaat uit volgende onderdelen: de eerste 2 bytes zijn vaste headers. Deze worden gevolgd door een byte die de lengte aangeeft. Hiermee wordt bedoeld hoeveel bytes mode, command en parameter samen zijn. Dit wordt gevolgd door eventuele parameters. En als laatste wordt een checksum mee verzonden. Een checksum is het resultaat van een bewerking op data. Deze wordt ook uitgevoerd door de ontvanger en vergeleken met de verzonden checksum. Indien deze niet gelijk zijn is er een overdrachtsfout veroorzaakt en wordt de boodschap niet verder verwerkt.

Tabel 6 Opbouw iPod boodschap

Naam	Grootte (bytes)	Waarde
Header	2	0xff 0x55
Length	1	Aantal bytes van (mode + command + parameter)
Mode	1	De mode van het commando
Command	2 of meer	Het commando
Parameter	0..n	Optionele parameter
Checksum	1	$0x100 - ((\text{Som van length/mode/command/parameter bytes}) \& 0xFF)$

Het protocol heeft 5 verschillende modes met bepaalde commando's

Tabel 7 Verschillende modes

Mode nummer	Doel
0x00	Mode 0 commando's worden gebruikt om van mode te veranderen.
0x01	Mode 1 wordt gebruikt voor spraak opname functies.
0x02	Mode 2 bevat de basis set functies. Dit zijn de commando's gebruikt om de iPod via een afstandbediening te besturen (play, pause, ...). Er zijn ook commando's om het iPod click <input type="checkbox"/> te simuleren. Deze commando's moeten echter wel gevolgd worden door een "key release commando".
0x03	Request Mode Status: Wanneer men naar mode 3 overschakelt, zet de iPod zich eigenlijk niet in een bepaalde mode maar geeft hij data terug waaruit af te leiden is in welke mode hij zich bevindt.
0x04	AiR Mode (Advanced iPod Remote). Bevat de uitgebreide commando's. Hiermee kan men gegevens opvragen van de iPod zoals de artiestnaam, titel, ... . In mode 4 worden de mode 2 commando's nog steeds aanvaard. Wanneer de iPod zich in deze mode zet zal hij het huidige nummer pauzeren en "ok to disconnect" op het scherm weergeven. Dit zijn enkele foutjes in deze mode.



We gebruiken echter enkel de modes 0, 2 en 4

Tabel 8 Mode 0

Commando	Uitleg
0x01 0x01	Ga naar Voice Recorder mode
0x01 0x02	Ga naar iPod Remote mode
0x01 0x04	Ga naar AIR mode

Tabel 9 Mode 2

Commando	Korte omschrijving	Uitleg
0x00 0x00	Key release	Dit commando wordt gebruikt om aan te geven dat de toets losgelaten is.
0x00 0x01	Play/Pauze	Als de iPod gepauzeerd is, dan zal deze beginnen spelen. Als de iPod aan het spelen is, dan zal de iPod pauzeren.
0x00 0x02	Volume +	Volume harder zetten
0x00 0x04	Volume -	Volume zachter zetten
0x00 0x08	Skip >	Vooruitspoelen naar volgend liedje
0x00 0x10	Skip <	Terugspoelen naar vorig liedje
0x00 0x20	Next Album	Volgend album afspelen
0x00 0x40	Previous Album	Vorig album afspelen
0x00 0x80	Stop	Muziek afspelen stoppen
0x00 0x00 0x01	Play (geen pauze)	Play
0x00 0x00 0x02	Pauze (geen play)	Pauze
0x00 0x00 0x04	Mute	Geluid afzetten
0x00 0x00 0x20	Next Playlist	Volgende afspeellijst afspelen
0x00 0x00 0x40	Previous Playlist	Vorige afspeellijst afspelen
0x00 0x00 0x80	Toggle shuffle	Shuffle aan- of uitzetten
0x00 0x00 0x00 0x04	iPod off	iPod opzetten
0x00 0x00 0x00 0x08	iPod on	iPod uitzetten
0x00 0x00 0x00 0x40	Menu Button	Menu knop indrukken
0x00 0x00 0x00 0x80	OK/ Select button	OK/select knop indrukken
0x00 0x00 0x00 0x00 0x01	Scroll up	Wiel simuleren
0x00 0x00 0x00 0x00 0x02	Scroll down	Wiel simuleren

Tabel 10 Mode 4

Commando	Parameter	Uitleg
0x00 0x14		Vraag iPod name
0x00 0x15	iPod Name as a null terminated string	
0x00 0x1C		Vraag lengte en status liedje
0x00 0x1D	lengte in <u>millisecond</u> (4 bytes) verstreken tijd in <u>ms</u> (4bytes)	
	status (1 byte): 0x00 stop, 0x01 playing, 0x02 paused	
0x00 0x1E		Vraag nummer van huidig liedje
0x00 0x1F	Nummer van huidig liedje (4 bytes)	
0x00 0x20	Nummer van liedje (4 bytes)	Vraag titel huidig liedje
0x00 0x21	Titel van huidig liedje ( <u>null terminated string</u> )	
0x00 0x22	Nummer van liedje (4 bytes)	Vraag artiest huidig liedje
0x00 0x23	Artiest van huidig liedje ( <u>null terminated string</u> )	
0x00 0x29	Commando	
	0x01 Play/pauze	
	0x02 Stop	
	0x03 <u>Skip</u> >	
	0x04 <u>Skip</u> <	
	0x05 <u>FFwd</u> = vooruit spoelen	
	0x06 <u>FRwd</u> = achteruit spoelen	
	0x07 Stop FF/RW	

Als er bij uitleg “vraag” staat, wil dit zeggen dat we dit commando zenden. De andere commando’s ontvangen we van de iPod.

## 14.15 SD-card

### 14.15.1 Inleiding



Figuur 56 SD-card

De Secure Digital card (SD-card) is de opvolger van het oudere type, de Multi Media Card (MMC). Het zijn beiden flashgeheugen kaarten. Ze zijn zeer populair in draagbare elektronica zoals digitale camera's, pda's, gsm, mp3-spelers... Het verschil tussen de SD-card en de MMC is dat de SD-card dikker is, maar een hogere schrijf- en leesnelheid heeft. De grootte van een SD-card is 32mm x 24mm x 2.1mm. De oudere MMC past nog wel in de SD-sleuven maar niet omgekeerd.

De naam “Secure” komt van de kopieerbeveiliging. Op een SD-card kan men bestanden opslaan die beveiligd zijn. Ze kunnen dan niet met een andere computer of apparaat gekopieerd worden.

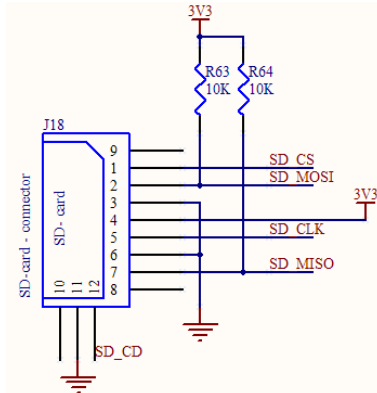
SD-kaarten zijn momenteel in de capaciteiten 128MB, 256MB, 512MB, 1GB, 2GB en 4GB verkrijgbaar. Ze zijn ontwikkeld om met een maximum snelheid van 25Mhz te werken. Dit is echter afhankelijk van de manier waarop de kaart aangesloten wordt. De 4GB kaarten worden ook SD High Capacity cards (SDHC) genoemd. Ze zijn alleen bruikbaar in de nieuwste generatie apparatuur met het SDHC logo.

Omdat de mogelijkheden met deze SD-kaarten zo uitgebreid zijn is besloten om hiervoor ook een connector te voorzien op dit development board.

### 14.15.2 Interface

De connector is aangesloten in SPI mode. Dit wil zeggen dat er een MOSI, MISO, CLOCK en een CS lijn gebruikt wordt. De SPI mode is reeds besproken bij het extra geheugen van de FPGA in hoofdstuk 1.3.2.1 .

Verder valt hier op te merken dat er een card detect is aangesloten. Dit is een optie van de connector. Wanneer er een kaart ingepluggd wordt zullen contact 11 en 12 inwendig met elkaar verbonden worden. Pin 12 is verbonden met een ingangspin van de FPGA zodat hiervan gebruik gemaakt kan worden om te weten of er een SD-card aanwezig is of niet.



Figuur 57 SD-card interface



Figuur 58 SD-card en connector

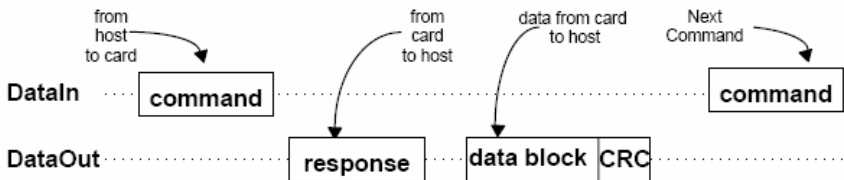
Tabel 11 Pinout SD-connector

Pin	Naam	Omschrijving	Pin	Naam	Omschrijving
1	CS	Chip Select	5	Sclk	Clock
2	DI	Data In	6	Vss2	Ground
3	VSS	Ground	7	DO	Data out
4	VDD	Voeding	8 & 9	-	-

### 14.15.3 Protocol

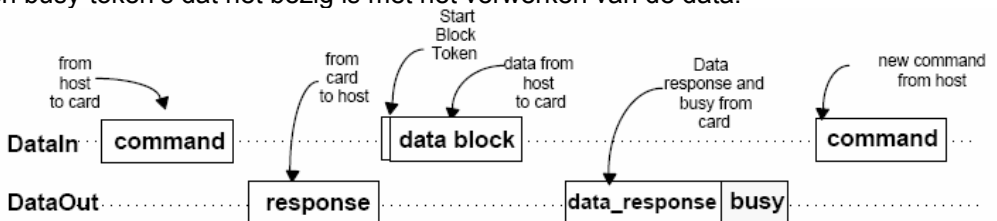
In SPI mode wordt er byte-georiënteerd gewerkt. Elk commando of datablok is opgebouwd uit 8 bits. De boodschappen bestaan uit commando's, responses (antwoorden) en datablokken. Elke transactie wordt gestart door de host die de CS lijn laag maakt.

Bij een leesinstructie wordt er door de host een commando gestuurd. De SD-card reageert hierop met een response token. Gevolgd door een data token met een 16bit-CRC (cyclic redundancy check). De maximum lengte van een data blok is 512 bytes. Er kan ook gebruik gemaakt worden van multi-block-reading. Hierbij blijft de SD-kaart datablokken versturen tot wanneer er een stop-commando door de host gestuurd wordt.



Figuur 59 lezen

Een schrijfinstructie kan ook weer in single- of multi-block. De host stuurt eerst een schrijfcommando. De SD kaart zal hierop reageren met een response token en zal wachten op een verzonden datablok. Na het ontvangen van dit blok zal de SD-kaart een data response token verzenden om aan te geven dat de data goed ontvangen zijn. De SD-kaart geeft ook aan d.m.v een busy-token's dat het bezig is met het verwerken van de data.



Figuur 60 schrijven

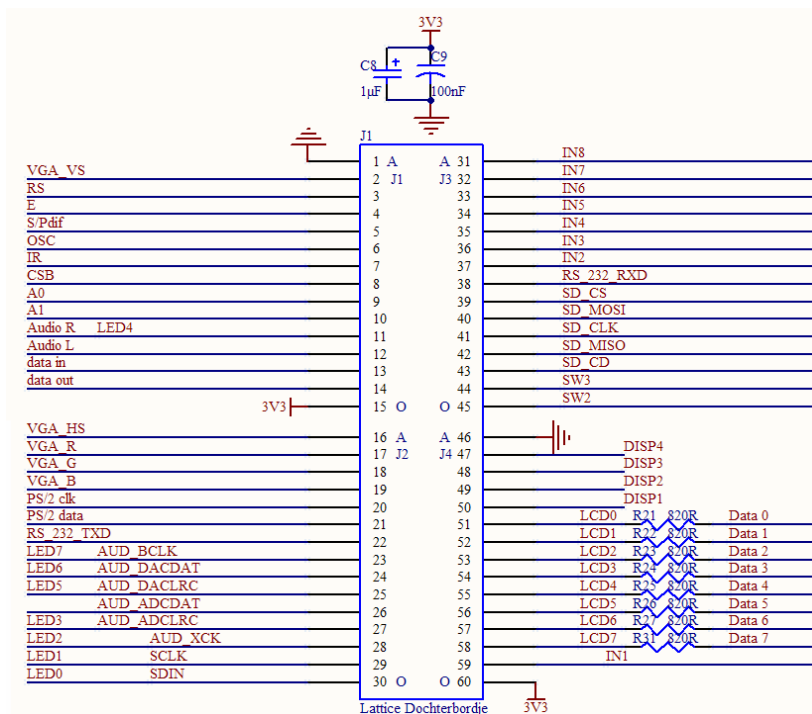
## 14.16 FPGA

Alles is gestructureerd naar de FPGA pinnen gebracht, zodat het geheel ook nog deftig getekend kan worden in de PCB editor.

Zoals gezegd werd er voor de FPGA een dochterbordje gebruikt. Hierdoor ligt er vast welke I/O pinnen er kan gebruikt worden.

Figuur 61 toont het schema met alle signalen van de interfaces naar de FPGA connector.

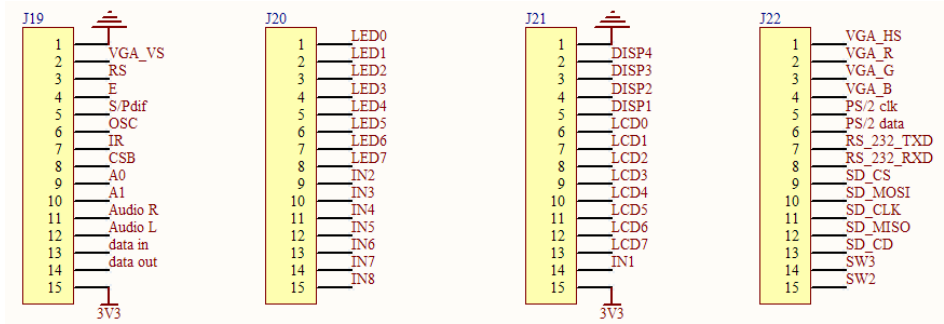
Een compleet overzicht van de I/O pinnummering is terug te vinden op het einde van dit hoofdstuk.



Figuur 61 FPGA Connector

Zoals in het schema te zien zijn er dus I/O pinnen met meerdere functionaliteiten zoals onder andere bij het LCD en de 7 segment displays. Hiervoor is er een extra dipswitch voorzien zodat deze afzonderlijk aan- of afgeschakeld kunnen worden.

Rond de FPGA connector zijn er ook nog footprints voor SIP female headers geplaatst. De headers zijn verbonden met de ernaast liggende I/O pinnen van de FPGA. Op deze manier zijn de signalen op de I/O pinnen gemakkelijk te meten met een oscilloscoop.



Figuur 62 Headers in schema

#### 14.17 Overzicht van de I/O pinnummering van de FPGA XP2-5

Dochterbordje nummering	Connector dochterbordje nummering	FPGA I/O	Signaal	Gemeenschappelijk signaal
J1A	1		GND	
J1B	2	13	VGA_VS	
J1C	3	17	RS	
J1D	4	22	E	
J1E	5	28	S/Pdif	
J1F	6	30	OSC	
J1G	7	40	IR	
J1H	8	43	CSB	
J1I	9	45	A0	
J1J	10	46	A1	
J1K	11	48	Audio R	
J1L	12	54	AudioL	
J1M	13	56	Data in	
J1N	14	58	Data out	
J1O	15		3V3	

J2A	16	11	VGA_HS	
J2B	17	15	VGA_R	
J2C	18	21	VGA_G	
J2D	19	27	VGA_B	
J2E	20	29	PS/2 CLK	
J2F	21	37	PS2/ DATA	
J2G	22	38	RS_232_TXD	
J2H	23	39	LED7	AUD_BCLK
J2I	24	44	LED6	AUD_DACDAT
J2J	25	47	LED5	AUD_DACLRC
J2K	26	53	LED4	AUD_ADCDAT
J2L	27	55	LED3	AUD_ADCLRC
J2M	28	57	LED2	AUD_WCK
J2N	29	61	LED1	SCLK
J2O	30	62	LED0	SDIN

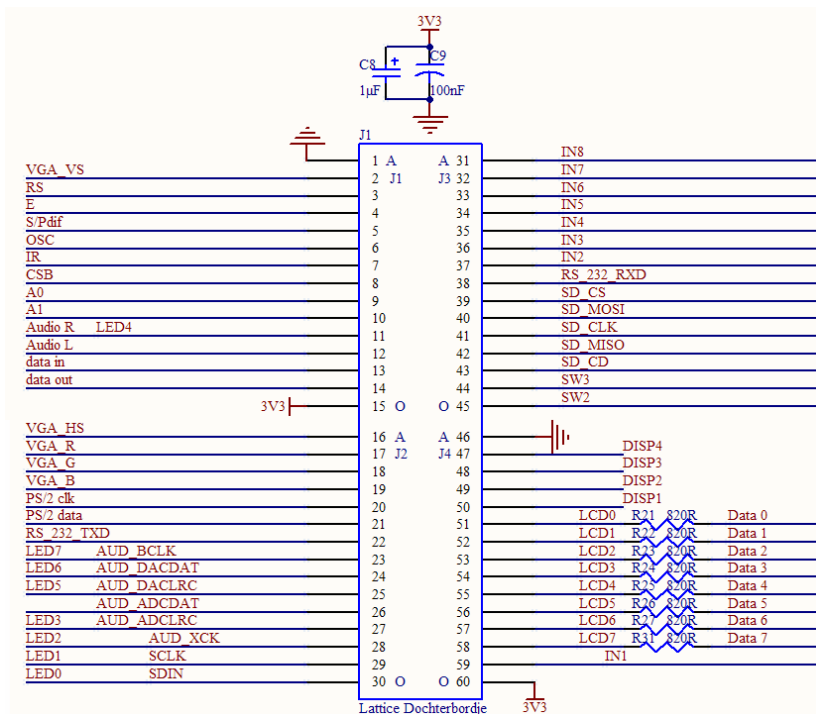
Dochterbordje nummering	Connector dochterbordje nummering	FPGA I/O	Signaal	Gemeenschappelijk signaal
J3A	31	87	IN8	
J3B	32	88	IN7	
J3C	33	90	IN6	
J3D	34	94	IN5	
J3E	35	102	IN4	
J3F	36	108	IN3	
J3G	37	110	IN2	
J3H	38	120	RS_232_RXD	
J3I	39	122	SD_CS	
J3J	40	124	SD_MOSI	
J3K	41	127	SD_CLK	
J3L	42	130	SD_MISO	
J3M	43	132	SD_CD	
J3N	44	134	SW3	
J3O	45	144	SW2	

J4A	46		GND	
J4B	47	89	DISP4	
J4C	48	92	DISP3	
J4D	49	100	DISP2	
J4E	50	107	DISP1	
J4F	51	109	LCD0	DATA0
J4G	52	119	LCD1	DATA1
J4H	53	121	LCD2	DATA2
J4I	54	123	LCD3	DATA3
J4J	55	125	LCD4	DATA4
J4K	56	129	LCD5	DATA5
J4L	57	131	LCD6	DATA6
K4M	58	133	LCD7	DATA7
J4N	59	142	IN1	
J4O	60		3V3	

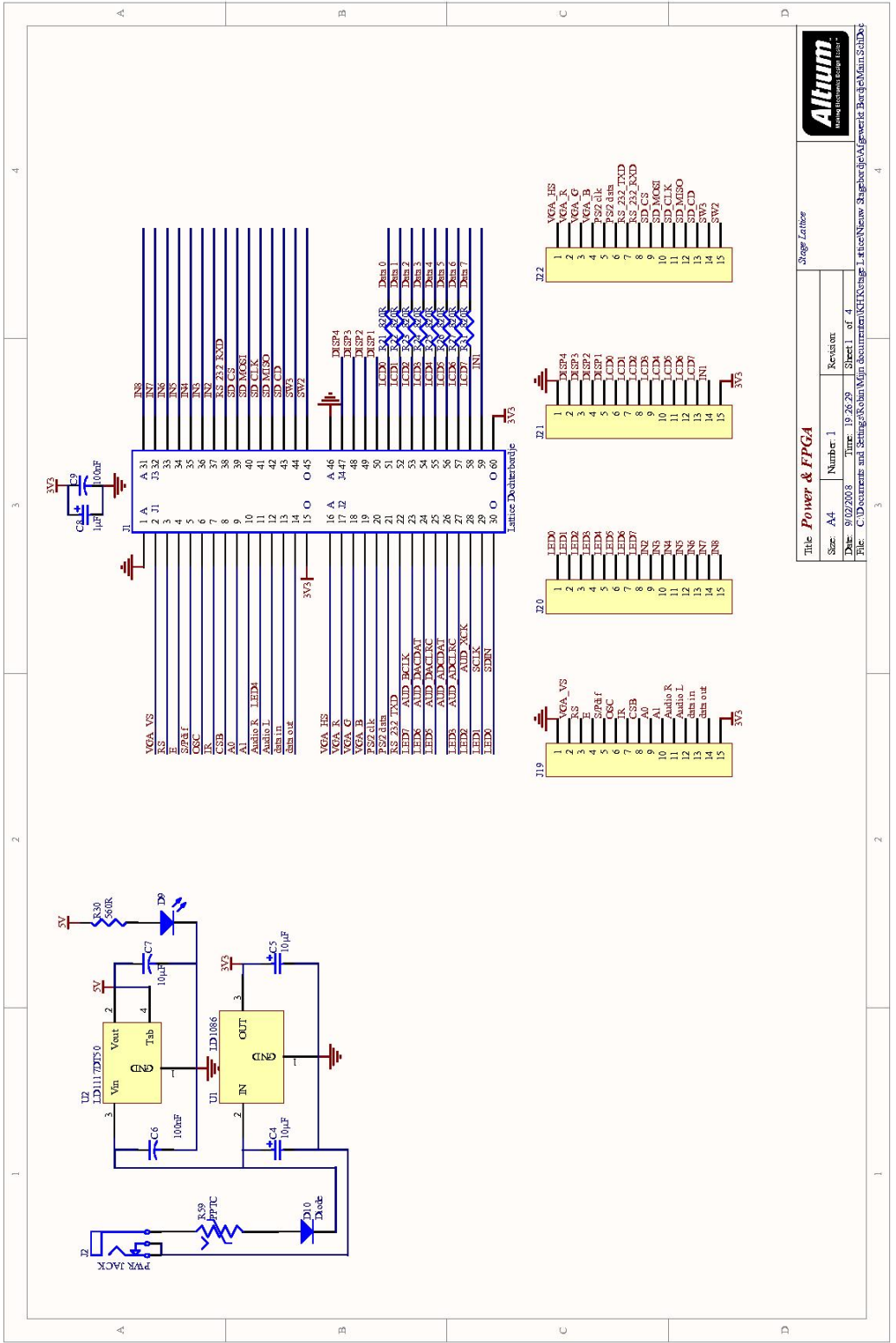


### 14.17.1 Verduidelijking van de kolommen:


- Dochterbordje nummering: De oorspronkelijke nummering van het beschikbare dochterbordje. Dit bestaat uit 4 connectors J1 - J4 met elk 15 pins benoemd van A tot O.
- Connector dochterbordje nummering: Dit is de nummering van de connector waar het eigelijke dochterbordje in geplugd wordt. Deze is ook schematisch afgebeeld in onderstaande afbeelding.
- FPGA I/O: dit is de pin die in de FPGA aangesproken moet worden.
- Signaal: Het signaal verbonden met die bepaalde pin:
- Gemeenschappelijk signaal: Bij sommige pins zijn er meerdere signalen bevestigd.



Figuur 63 FPGA connector

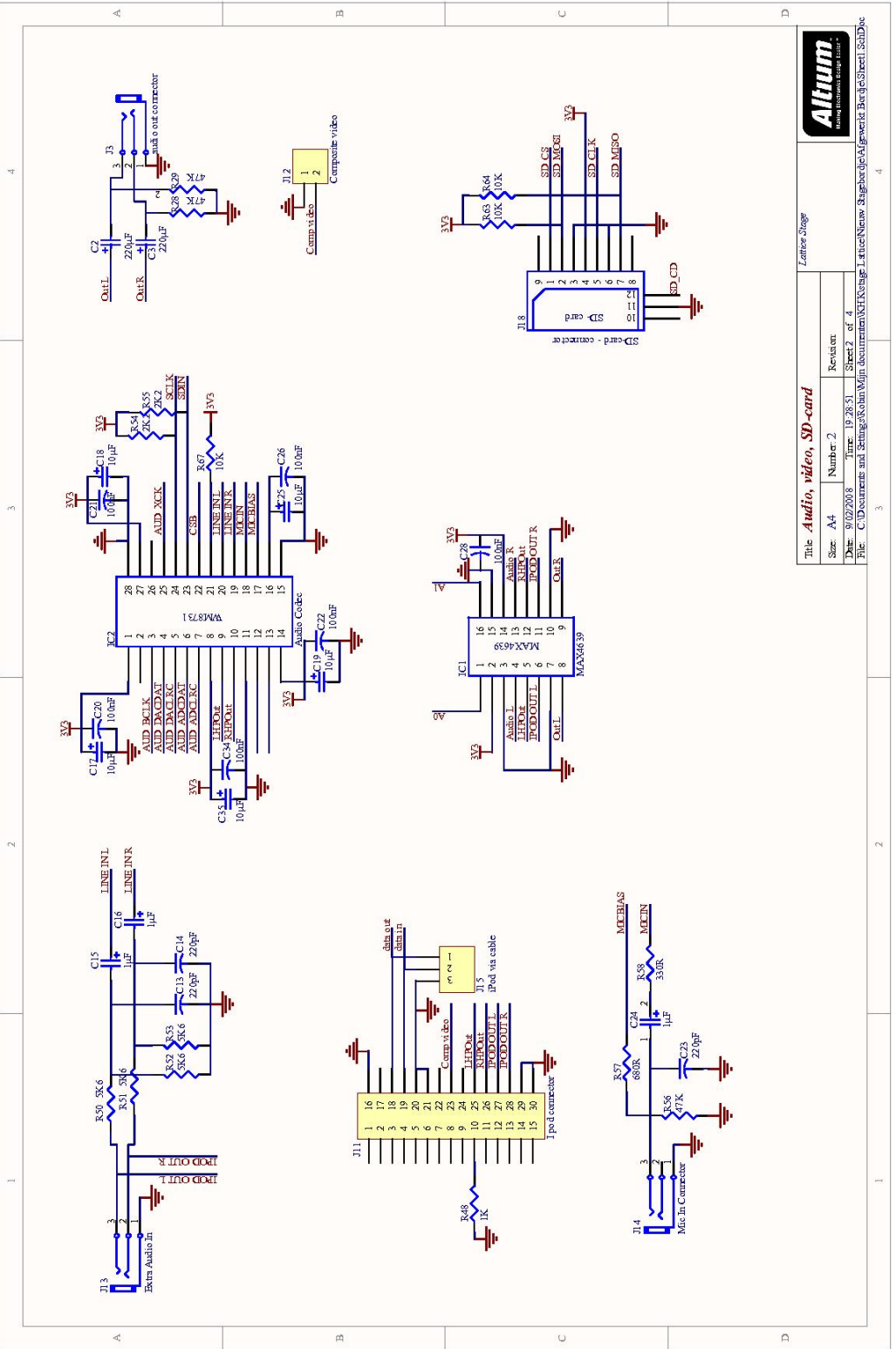


Hoofdstuk 14 Lattice development board LFXP2-5E - 58 -


  
 Altium  
 www.altium.com

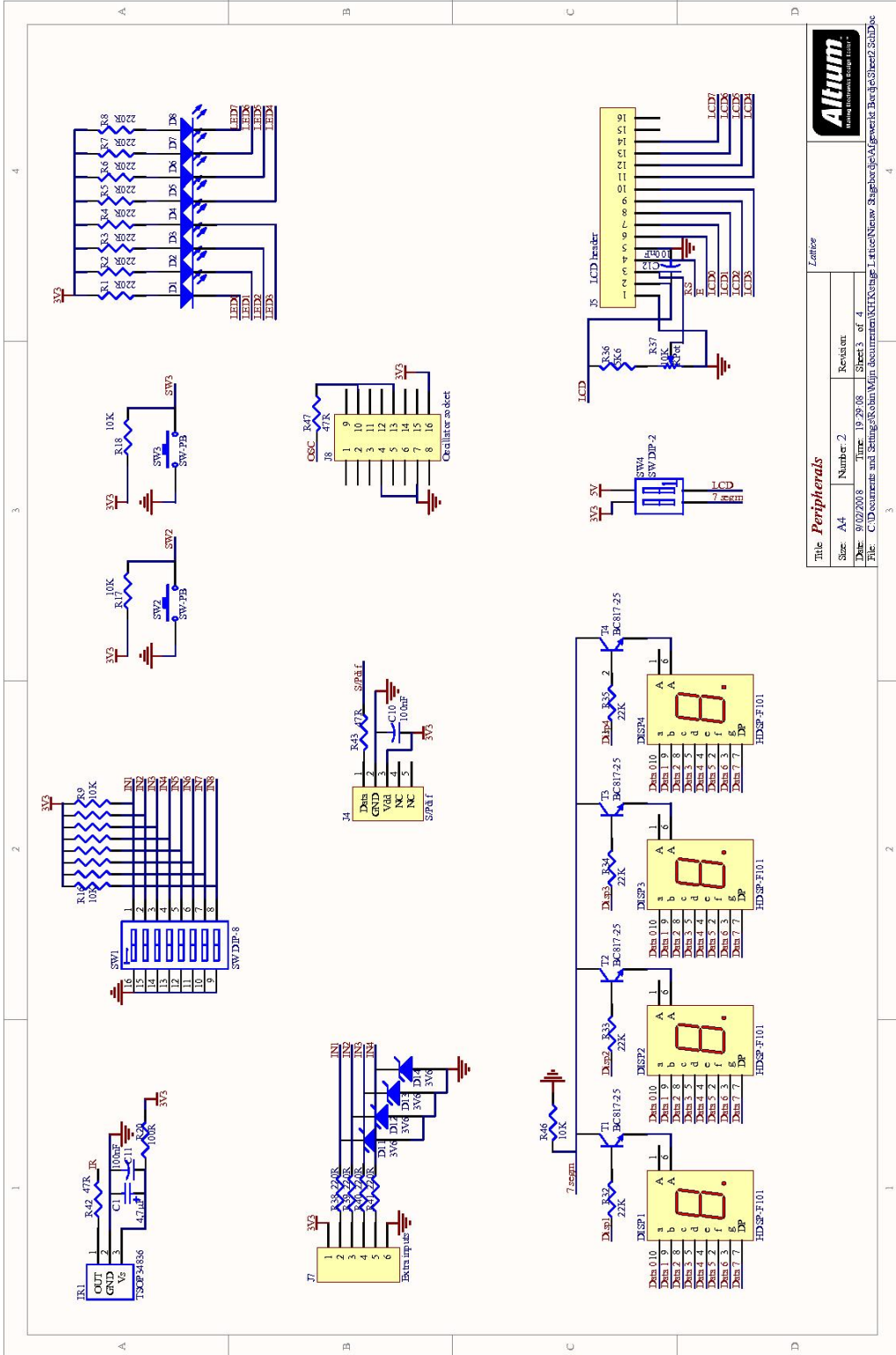
Title **Power & FPGA** Stage Lattice

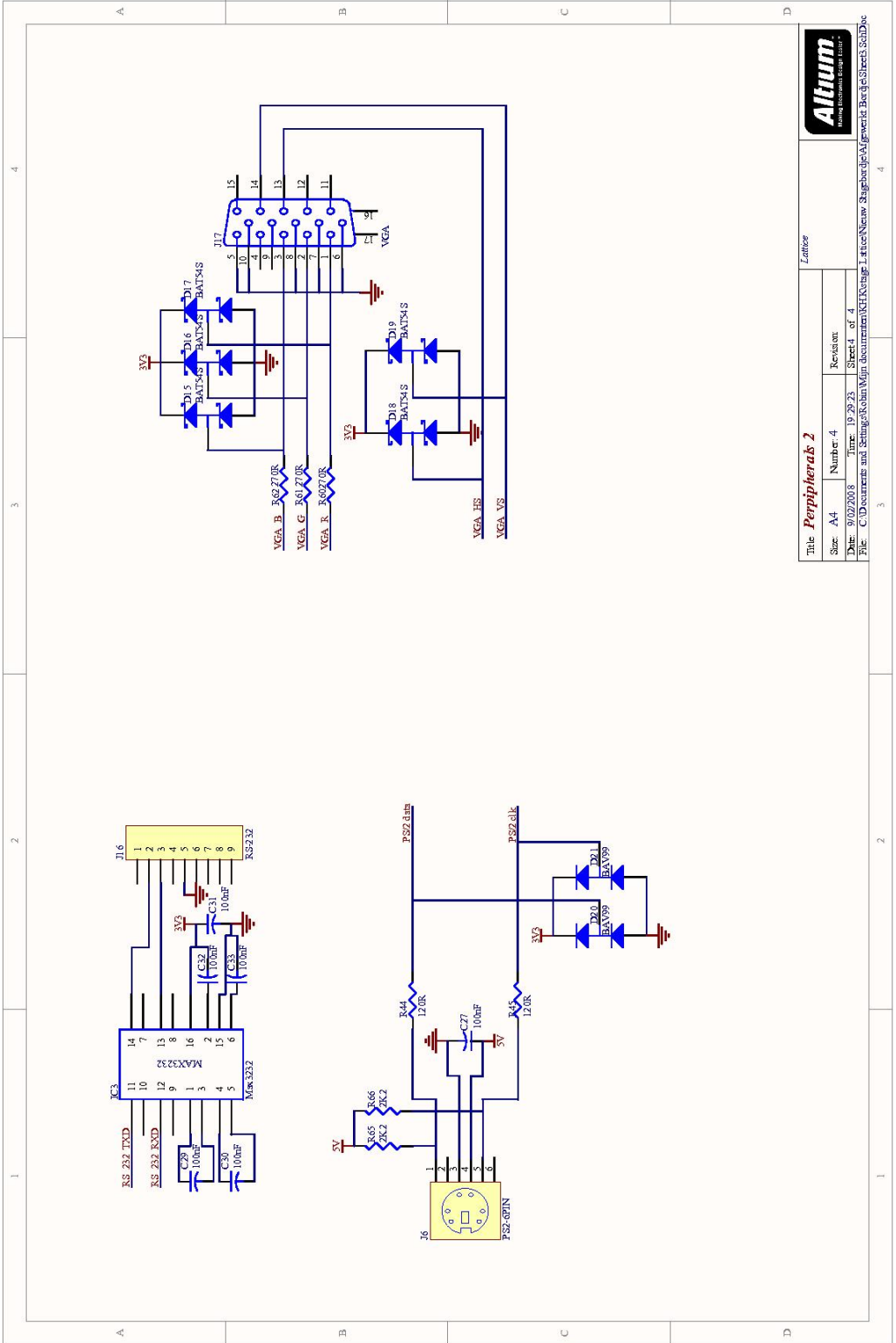
Size: A4	Number: 1	Revision:	Sheet 1 of 4
Date: 9/02/2008	Time: 19:26:29		
File: C:\Documents and Settings\kohn\My Documents\KHN\Stage 1.rtf\New_Stage\Board\Main_SchDoc1			



The **Audio, video, SD-card** Lattice design

Size: A4	Number: 2	Revision:
Date: 9/02/2008	Time: 19:28:51	Sheet: 2 of 4
File: C:\Documents and Settings\Kobin\My Documents\NGH\Kulagge_14\New\New_Sapphire\Algoworld\BoardSheet1_SchDoc		





Title <b>Peripherals 2</b>		Revision	
Size: A4	Number: 4	Revision:	
Date: 9/02/2008	Time: 19:29:25	Sheet 4 of 4	
File: C:\Documents and Settings\Robin Mijm\Documents\GHE\Kitsge-14\New\New_Schapsche\Algoworld\BoardSheet\SD.Dwg			

