Labo 1: Vivado Tutorial + aanvullingen

Door: Arvid Van Dorpe, Mike Molengo

1. Inleiding

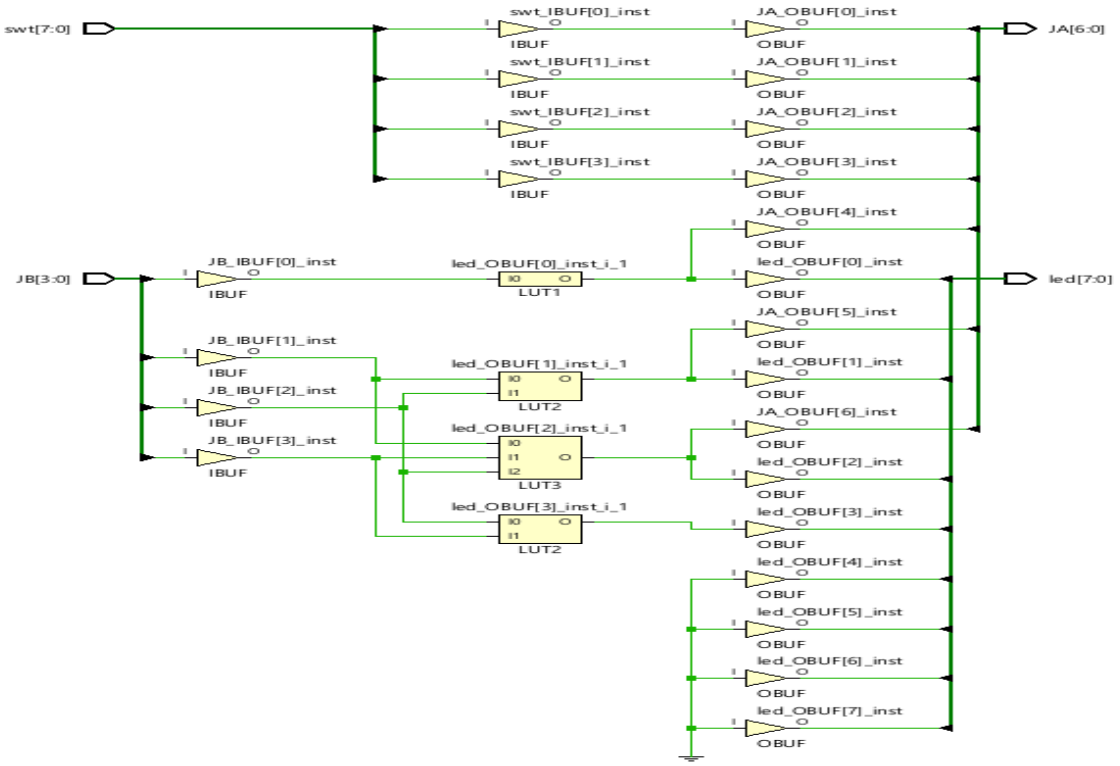
In dit labo werd gewerkt met de Xilinx Vivado Software om een digitaal ontwerp te implementeren en testen op een FPGA-bord. De doelstelling van het labo was om een functioneel ontwerp te realiseren en dit te verifiëren door middel van simulaties en metingen met een logic analyzer en oscilloscoop. Dit verslag documenteert de verschillende fasen van het labo, inclusief de functionele analyse, RTL-ontwerp, teststrategie en implementatie op het Basys 3-bord.

2. Opdrachten/vereisten

Het ontwerp bestond uit een eenvoudige digitale schakeling waarbij bepaalde invoersignalen (switches of functiegenerator) direct op de uitvoer-LEDs weergegeven werden, terwijl andere invoersignalen werden verwerkt via logische bewerkingen voordat ze werden weergegeven. De implementatie gebeurde op een Basys 3-FPGA-bord (Artix 7).

3. RTL-schema en VHDL-code

Het RTL-schema van het ontwerp werd gegenereerd met Vivado en toont de logische componenten en verbindingen binnen het circuit



Bovenstaande schema toont het schema aan voor de laatste opdracht in dit labo. De PMOD JB-poorten worden gebruikt om de 4 bit-output van de functiegenerator binnen te nemen. Deze vervangen de switches 4 tot 0 van het origineel schema.

De VHDL-code bevat de logische implementatie en kan hieronder worden geraadpleegd. Dit is ook de code voor het laatste deel van het labo.

```
    port (
        swt : in STD_LOGIC_VECTOR(7 downto 0);    -- Ingang schakelaars
        led : out STD_LOGIC_VECTOR(7 downto 0);   -- Uitgang LEDs
        JA  : out STD_LOGIC_VECTOR(6 downto 0);    -- Uitgang naar PMOD JA[6:0]
        JB : in std_logic_vector(3 downto 0)

    );
end tutorial;

Architecture behavior of tutorial Is

    Signal led_int : STD_LOGIC_VECTOR(7 downto 0) := "00000000";

    begin
        led <= led_int;  -- LEDs koppelen aan interne signalen

        -- LED-logica met input functie
        led_int(0) <= not(JB(0)); --inverteer JB0
        led_int(1) <= JB(1) and not(JB(2)); -- and poort
        led_int(3) <= JB(2) and JB(3); -- and poort
        led_int(2) <= led_int(1) or led_int(3); -- or

        -- LED-logica met switches
        --led_int(0) <= not(swt(0));
        --led_int(1) <= swt(1) and not(swt(2));
        --led_int(3) <= swt(2) and swt(3);
        --led_int(2) <= led_int(1) or led_int(3);
        --led_int(7 downto 4) <= swt(7 downto 4);

        -- JA PMOD-uitgangen koppelen
        JA(0) <= swt(0);       -- Directe koppeling van schakelaars
        JA(1) <= swt(1);
        JA(2) <= swt(2);
        JA(3) <= swt(3);
        JA(4) <= led_int(0); -- Output van LED-logica naar PMOD
        JA(5) <= led_int(1);
        JA(6) <= led_int(2);
```

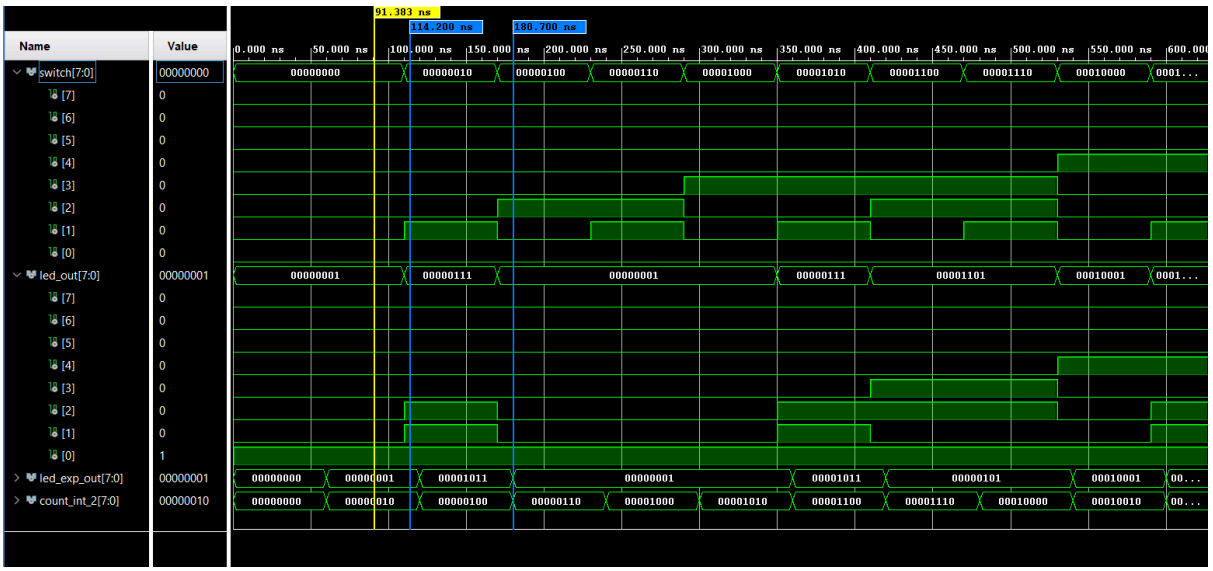
4. Implementatie en constraints

Het ontwerp werd gesynthetiseerd en geïmplementeerd in Vivado, waarbij gebruik werd gemaakt van een aangepaste XDC-file voor correcte pin-mapping.

5. Testen en validatie

5.1 Simulatie

De eerste validatie werd uitgevoerd met Vivado XSIM, waarbij de verwachte functionaliteit werd geverifieerd.



5.2 Hardware-metingen

Na de implementatie werd de bitstream gegenereerd en op het FPGA-bord geladen. Tests met fysieke schakelaars en LEDS (aanwezig op het bord) bevestigden de correcte werking.

5.2.1 Foto 1

Alle schakelaars staan in de lage stand, en LED0 brandt. Dit klopt, aangezien LED0 is aangesloten op een NOT-poort met als ingang switch 0.

5.2.2 Foto 2

Switch 1 staat hoog. Hierdoor branden LED0, LED1 en LED2. Dit is logisch, want:

- LED1 is de uitgang van een AND-poort-switch 1 en Not-switch 2 als ingangen.
- LED2 brandt omdat het de uitgang is van een OR-poort met als ingangen LED1 en LED3.

5.2.3 Foto 3

Switch 1 en switch 2 staan hoog. In deze situatie brandt alleen LED0.

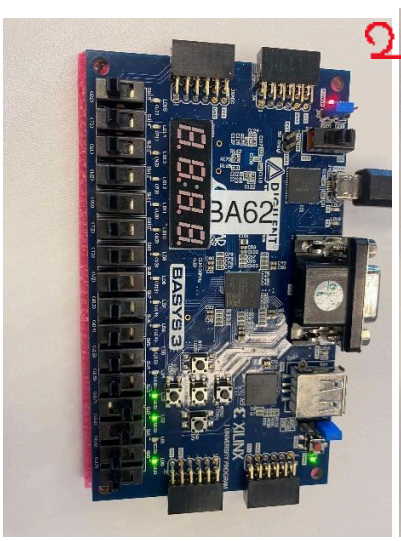
- LED1 blijft uit omdat het de uitgang is van een AND-poort met als ingangen switch 1 en NOT switch 2. Aangezien switch 2 hoog is, levert de NOT-poort een 0, waardoor de AND-poort ook 0 weergeeft.
- LED2 blijft uit, aangezien het de uitgang is van een OR-poort met als ingangen LED1 en LED3. Omdat zowel LED1 als LED3 uit zijn, blijft LED2 ook uit.

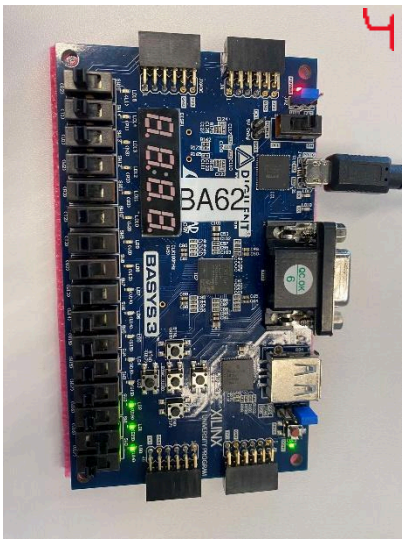
5.2.4 Foto 4

Switch 2 en Switch 3 staan hoog. In deze situatie branden LED0, LED2, LED3.

- LED0 brandt vanwege switch 0, die laag staat en verbonden is met een NOT poort.
- LED3 brandt omdat het de uitgang is van een AND-poort met switch 2 en switch 3 als ingangen.
- LED2 brandt omdat het de uitgang is van een OR-poort met LED1 en LED3 als ingangen. Aangezien LED3 aanstaat zal LED2 ook branden.

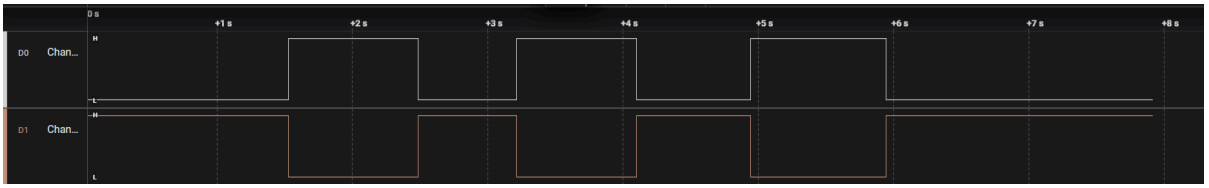
Foto	Swt0	Swt1	Swt2	Swt3	LED0	LED1	LED2	LED3
1	0	0	0	0	1	0	0	0
2	0	1	0	0	1	1	1	0
3	0	1	1	0	1	0	0	0
4	0	0	1	1	1	0	1	1



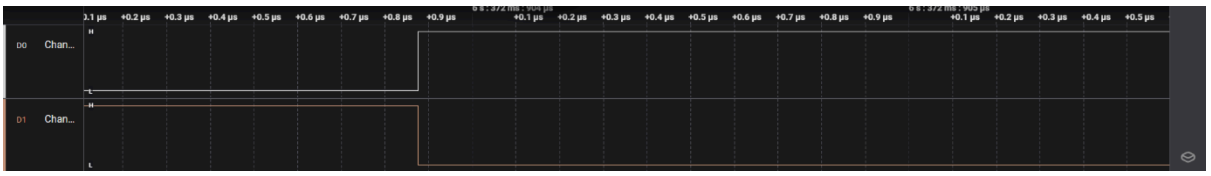


5.3 Metingen met logic analyser

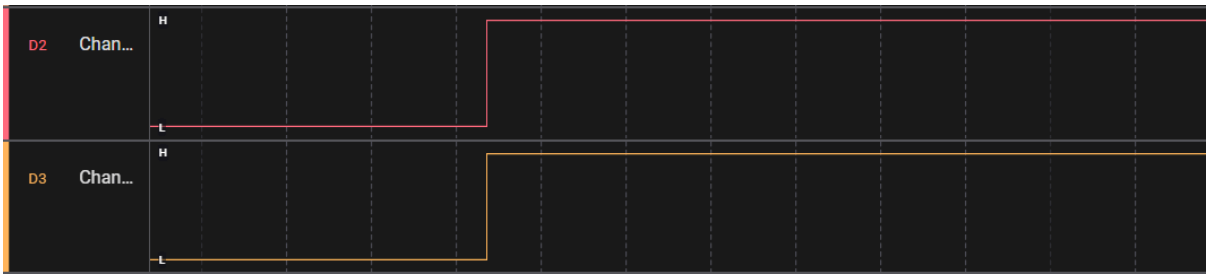
Op de foto hieronder is te zien hoe switch 0 en LED0 op elkaar reageren. Dit is een uitgezoomde versie.



Hieronder staat een meer ingezoomde versie van de meting. Zoals je kan zien is er niet veel vertraging tussen invoer en uitvoer. Dit wordt later uitgelegd.



Hieronder is een foto te zien waar switch 1 drie keer aan- en uitgezet wordt. Hier is ook weer heel weinig tot geen vertraging tussen invoer en uitvoer te zien.



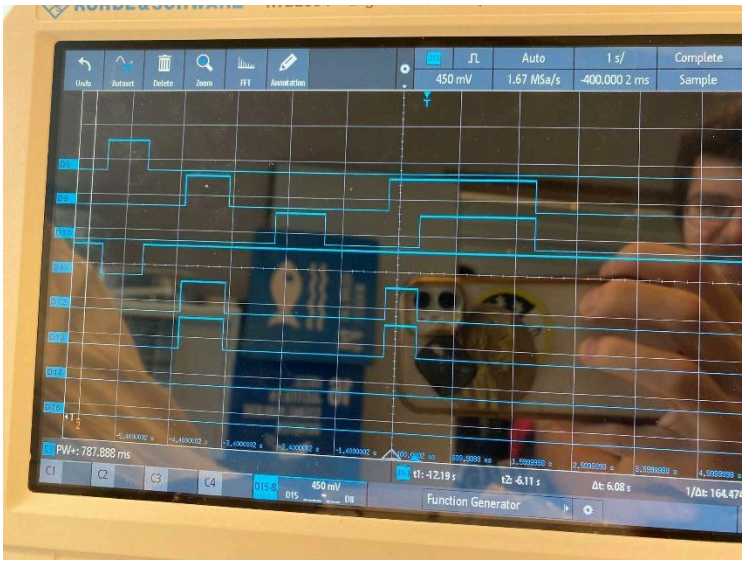
Hieronder is een foto te zien van hoe switch 1 en 2 met elkaar werken. Ook zijn LED1 en 2 te zien.

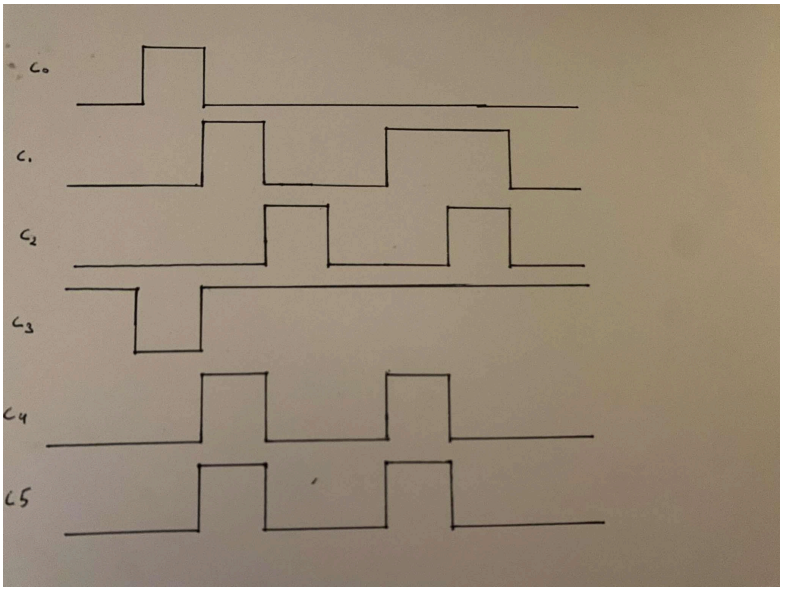
Waarom is er geen vertraging te zien op deze scopebeelden?

De logic analyser heeft een maximale sample rate van 24 MHz. Dit betekent dat hij elke 41.67 ns een meting doet. De vertraging die verwacht wordt is kleiner dan dit tijdsinterval. Hierdoor kan de Logic Analyzer het dus niet goed waarnemen. Tijdens het labo hadden we voor één keer geluk waardoor we het verschil ietswat konden zien. Maar dit was niet betrouwbaar te repliceren.

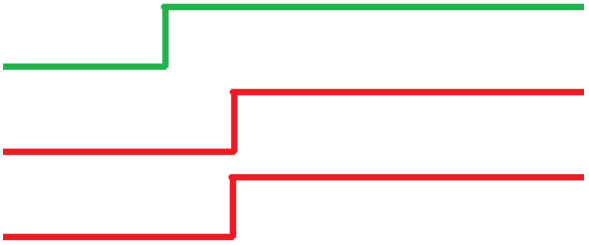
5.4 Meting met oscilloscope

1. **Algemene werking**
   Dit beeld laat de interactie zien tussen switch 0,1 en 2 en de bijbehorende LED's (LED 0,1 en 2). Het toont hoe de schakelaars invloed hebben op de LED's via de aangesloten logische poorten.

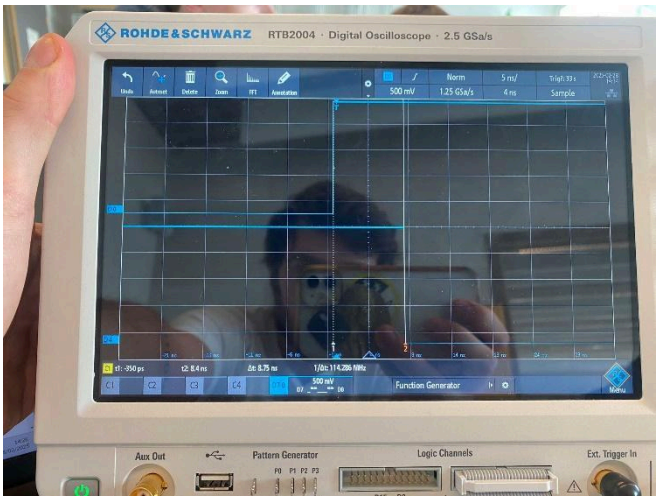




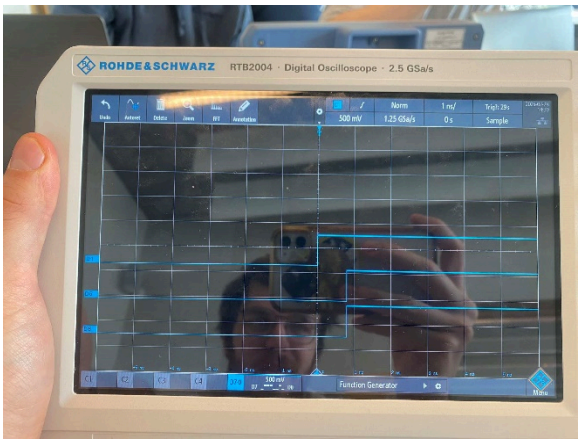
2. **Werking en delay van switch 1**
   Hier is te zien hoe switch 1 wordt geactiveerd en vervolgens LED1 en LED2 inschakelen. De gemeten vertraging via de oscilloscoop is 800 ps.



3. **Werking en vertraging van switch0 en LED0**
   Dit beeld toont specifiek de relatie tussen switch 0 en LED0.
   De gemeten vertraging hier bedraagt 8.75 ns.



4. **Werking en delay van switch 2 en 3**
   Dit beeld laat zien hoe switch 2 en switch 3 samenwerkingen om LED 2 in te schakelen. De gemeten vertraging is hier 850 ps.



5. Logische poorten en verbindingen

- Switch 0 □ NOT-poort □ LED0
- Switch 1 & NOT switch 2 □ AND Poort □ LED1
- LED1 & LED3 □ OR-poort □ LED2
- Switch 2 & switch 3 □ AND poort □ LED3

De oscilloscoop metingen hierboven bevestigen de correcte werking van de schakelingen en tonen de vertragingen bij het schakelen van de LEDs.

6. Conclusie

Het ontwerp functioneerde zoals verwacht, zowel in simulatie als op hardware. De teststrategie bood een grondige validatie en de oscilloscoop metingen gaven extra inzicht in het tijdsgedrag van het circuit

7. Gebruikte bronnen

Bron	Beschrijving
Vivado Tutorial	Handleiding voor Vivado-software
Xilinx documentatie	FPGA-specificaties en constraints