Rekenaarstelsels 245 - Prakties 10
Computer Systems 245 - Practical 10

2015-10-16

1 Opdrag 1

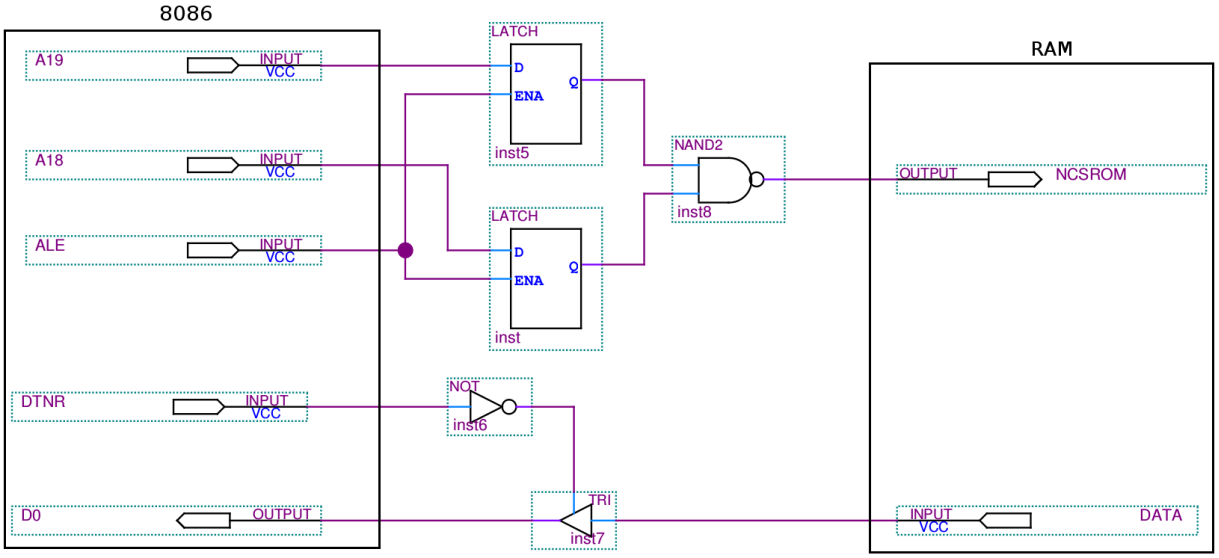
Hierdie opdrag gebruik die **Quartus II 9.0sp2 Web Edition** ontwikkellingsomgewing.

Skep 'n nuwe Quartus projek en gebruik 'n toestel uit die Stratix II reeks, die EP2S15F484C3. Skep 'n nuwe **Block Diagram File** en **Vector waveform File** vir die projek. Teken die blokdiagram en stel die golfvorm op soos dit hieronder verskyn. Die diagram wys 'n gedeelte van die geheue-koppelvlak van 'n 8086 mikroverwerker. Die komponente wat ingesluit is is die adreslyne (A18 en A19), die adresgrendel (ALE), 'n geheueselekteerlyn (NCSROM) en een Datalyn (D0).

1 Assignment 1

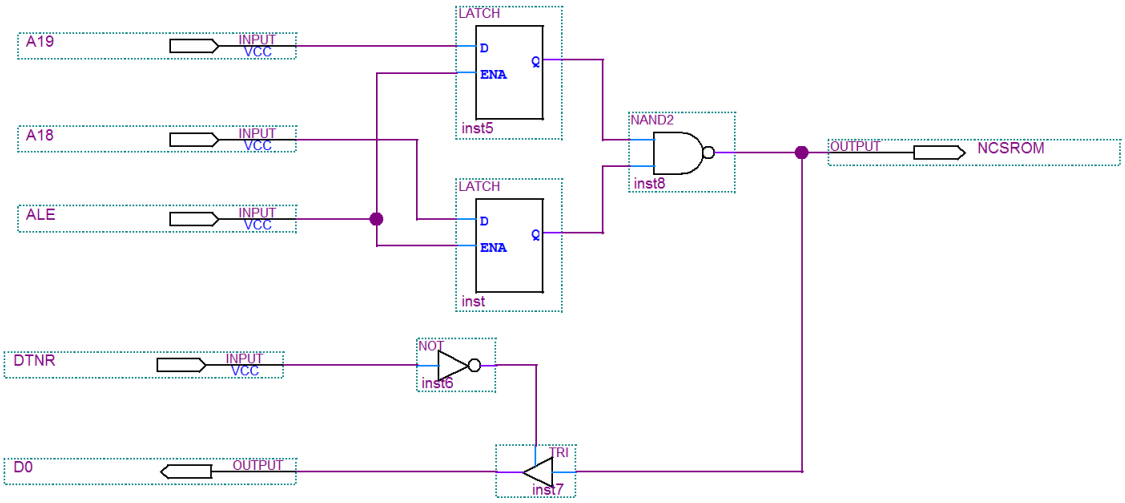
This assignment makes use of the **Quartus II 9.0sp2 Web Edition** development environment.

Create a new Quartus project using a Stratix II device, the EP2S15F484C3. Create a new **Block Diagram file** and a new **Vector Waveform file** for this project. Draw the block diagram and set up the vector waveform file exactly as shown below. The diagram shows a portion of the memory interface to an 8086 microprocessor. The components included are two address lines (A18, A19), the Address Latch Enable (ALE) line, one memory select line (NCSROM) and finally one data line (D0).



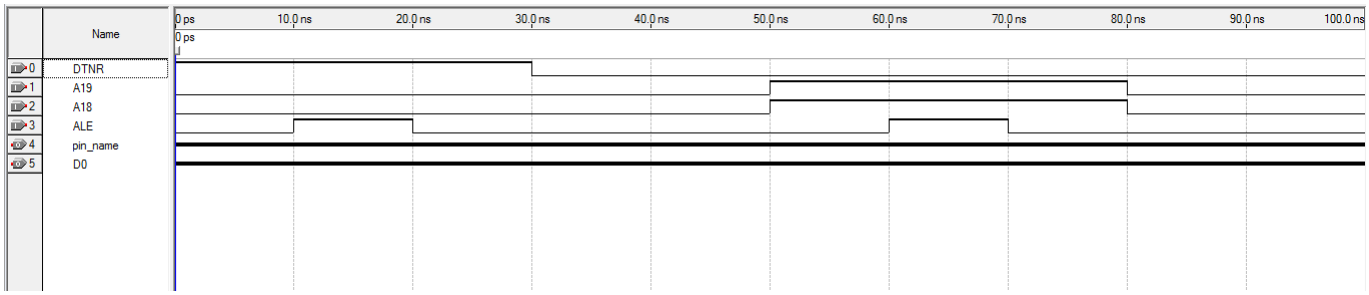
Om RAM uitree te simuleer is die data uitree verbind aan die geheueselekteerlyn. Bou die stroombaan in Quartus soos hieronder gewys.

To simulate RAM output, the data output is connected to the memory select line. Build the circuit as follows in Quartus.



Stel die project saam en simuleer dit. Let op dat 'n **Timing** simulasie gedoen moet word, nie 'n funksionele simulasie nie. Om dit te verander, gaan na *Assignment, Settings, Simulator Settings* en maak seker dat die simulasie modus op **Timing** is. In die simulasie word aanvaar dat die ROM vertraging zero is (die uitree verander onmiddellik as die geheue selekteer lyn verander).

Compile the project and run a simulation. Note that a **Timing** simulation must be performed, not a Functional simulation. To change this, go to Assignment, Settings, Simulator Settings, and check that the simulation mode is set to timing. In the simulation the ROM delay is assumed to be zero (the output changes immediately the chip select changes).



Gebruik die simulator resultate, en meet die vertraging van die volgende komponente.

1. Tristate grendel
2. Latch en NAND Gate kombinasie

Using the simulator results, measure the delays of the following components

1. Tristate Latch
2. Latch and NAND Gate combination

Wat is die maksimum vertraging wat die ROM kan hê vir 'n 8086 sisteem wat loop teen 'n klokfrekwensie van 10MHz, as die gegewe komponente gebruik word, en die vertragings soos volg is? Toon jou berekeninge. Verwys na lesing 23 vir meer inligting.

What is the maximum delay that the ROM may have for an 8086 system running at a clock frequency of 10MHz if the simulated components are used, and the delays are a follows? Show your calculations. Refer to lecture 23 for more information.

$$T_{CLAVmax} = 50ns$$

$$T_{DVCLmin} = 5ns$$

Ons gaan nou dit vergelyk met twee ander toestelle. Gaan na *Assignments* ⇒ *Device*, verander die toestel en stel die projek weer saam. Herhaal die vorige metings vir die

We are now going to compare this to two other devices. To change the device go to *Assignments* ⇒ *Device*, change your device, and recompile the project. Repeat the previ-

nuwe toestel. Toon jou antwoorde in ‘n tabel.

- Stratix II EP2S15F484C5
- Cyclone III EP3C5E144C8

2 Opdrag 2

Pas laasweek se e^{2}studio projek aan om die volgende te doen:

1. Wanneer die letter ’a’ ontvang word, stuur die waarde terug wat die ADC lees vanaf die intree wat aan die verstelbare weerstand op die demobord gekoppel is.

2. Verander die “interval timer” om ‘n teller te inkrementeer elke sekonde. Wanneer die letter ’b’ op die UART ontvang word, stuur die waarde van die teller terug. Jou vorige timer opdrag moet nog steeds werk.

Om vanaf die ADC te lees, stel die *Peripheral Function* op soos hieronder. Onthou om die kode weer te genereer.

ous measurements. Present your findings in a table.

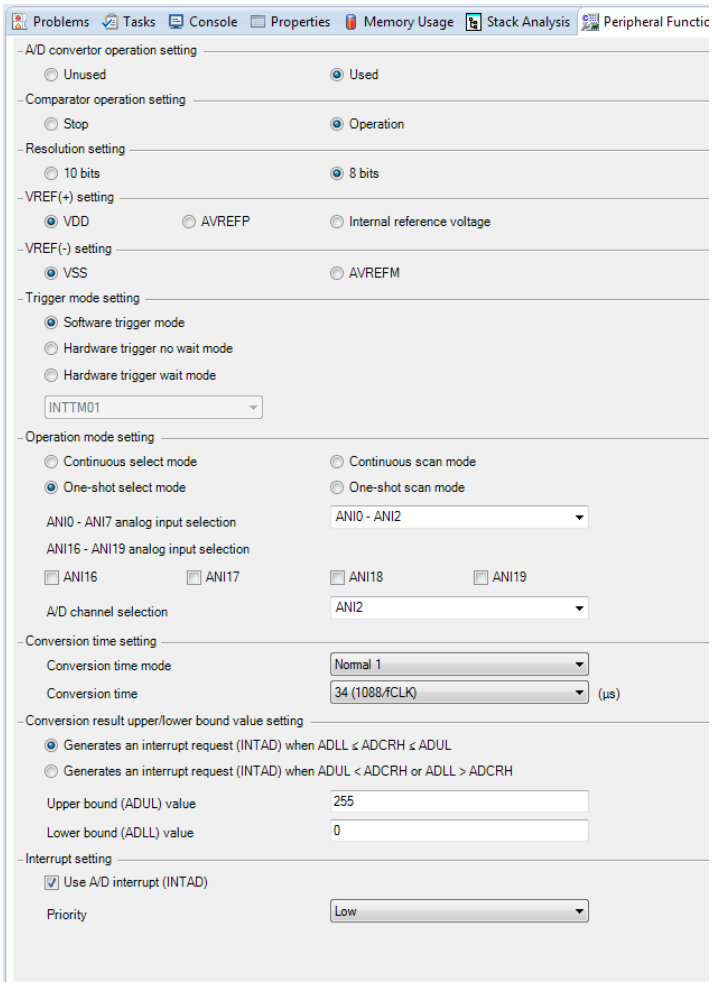
2 Assignment 2

Modify last weeks e^{2}studio project to do the following:

1. When the character ’a’ is received, send the value that the ADC is reading from the adjustable resistor on the demo board.

2. Modify the interval timer to increment a counter every second. When the character b is received on the UART, send the value of the counter back. You previous timer operations must still work.

To read from the ADC, set up the peripheral function as shown below. Remember to click “generate code”.



Die kode om van die ADC te lees is soos volg:

```
//To setup the ADC
R_ADC_Create();
R_ADC_Start();

//To Read a value
uint8_t adcvalue=0;
R_ADC_Start();
while(adc_done_flag==0); //Wait for the interrupt
adc_done_flag=0;
R_ADC_Get_Result_8bit(&adcvalue);
```

Die `adc_done_flag` moet in die *callback*-funksie van die ADC gestel word.

3 Inhandiging inligting

Handig opdrag 1 in op `learn.sun.ac.za` voor **17:00 vandag**. Jou inhandiging moet 'n PDF-lêer wees.

Handing opdrag 2 in op `learn.sun.ac.za` voor **23:55 op Donderdag 22 Oktober**. Jou inhandiging moet 'n zip-lêer van die hele projek wees.

The code to read from the ADC:

```
//To setup the ADC
R_ADC_Create();
R_ADC_Start();

//To Read a value
uint8_t adcvalue=0;
R_ADC_Start();
while(adc_done_flag==0); //Wait for the interrupt
adc_done_flag=0;
R_ADC_Get_Result_8bit(&adcvalue);
```

The `adc_done_flag` needs to be set in the ADC callback function.

3 Submission information

Hand in assignment 1 on `learn.sun.ac.za` before **17:00 today**. Your submission should be a single PDF.

Hand in assignment 2 on `learn.sun.ac.za` before **23:55 on Thursday 22 October**. Your submission should be a zip-file of the entire project.