Executive Summary

Fujitsu Semiconductor Europe is a design centre for high speed analogue to digital convertors (ADCs) and digital to analogue converters (DACs) for communication applications in Maidenhead, Berkshire. My placement has been based in the applications department, which involves the design of customer development boards and on-going customer support for products. During the placement, I have covered a number of engineering disciplines including user interface design, digital design and PCB design.

My first major project involves designing a MATLAB GUI (graphical user interface) to automatically generate the test vector needed at the input of the DAC during testing in the lab. This requires an understanding of the parameters needed to create test vectors, which will either be a time domain sine wave or a communication standard waveform such as UMTS. It is also important to know the effect sampling rate and FIR filters built into the current DAC product, will have on the output vector. The GUI needs to be able to give users information about these limitations, for example, a warning message if a sine wave frequency is less than the Nyquist frequency of the system. The vectors generated by the GUI will need to be checked in the lab using the DAC development board and a frequency spectrum analyser.

My second project involves sourcing a replacement for an ADC (different from the main Fujitsu ADC) on a development board, which is used to process signals on the board. A small milled board can be created to test the component and signals can be wired from test points on the main board to pins on the milled board. A schematic and PCB (printed circuit board) layout needed to be created for the board which can then be milled and have components soldered on in the company labs. The FPGA (field programmable gate array) control code, which contains the digital control and timing information, also needs updated for the new ADC. Finally MATLAB scripts used to send data to the board need reviewed to send the correct data to the FPGA and ensure the data is read back in the correct format.

My current project centres on digital design, a new development board (2) has space to use either an ADC or a DAC whereas the previous version of the development board contains only space to use an ADC. This means the control code in the FPGA needs to be updated to send control signals to the selected device only (ADC or DAC) and set the pins on the other device into high impedance mode. This project includes writing a new VHDL control script for a microcontroller which can control power supplies to the board. Finally this will require writing a MATLAB function to send the data to the development board.

Contents

Introduction

Fujitsu Semiconductor Europe is a design centre for high speed analogue to digital converters (ADCs) and digital to analogue converters (DACs) for communication applications based in Maidenhead, Berkshire. Products recently released by the department include a 12GSPS digital to analogue converter.

There is a team of around 50 engineers on site working in three main areas: design, applications and test and evaluation. Design involves the transistor level design of the application specific integrated circuit (ASIC) and is divided into the analogue and digital teams. Applications department involves the design of development boards for customers including the design and layout of board and the digital interfaces needed to communicate with the ADC and DACs. It also provides some ongoing technical support to customers. The test and evaluation department tests new devices to ensure that they meet specifications and identifies problems within devices in the engineering phase of their development.

My placement has been undertaken in the applications department and has covered various different engineering areas. These include designing a MATLAB user interface to improve test processes and involved good knowledge of sampling theorem and filters, also replacing a part on a development board which covered printed circuit board (PCB) design and understanding VHDL (a hardware description language) to adapt a serial peripheral interface (SPI). Other projects have built on these skills such as producing small PCBs in order to test circuits which will be used on the main development board and using digital design to create SPI interfaces to extend the control capabilities of current development boards.

Vector Generator User Interface

Context

To test digital to analogue converters (DACs), a test vector must be applied at the input. This vector contains a sequence of numbers ranging between 0 and 16383 which can be arranged to produce a sine wave output or a wideband output, such as the UMTS (Universal Mobile Telecommunications System) waveform.

The historic approach to producing these vectors is to alter constants, such as sampling frequency or vector length, directly in a MATLAB script then run the script to produce the desired vector. This approach is limited as it required the user to have good knowledge of MATLAB and the required constants are not clearly presented within the MATLAB code. Also, the scripts do not carry out any validation checks on the constants entered and a unique name for each saved vector is not generated thus leading to a potential problem, as two different vectors could be created with the same name.

The project is relevant to MEng Electrical and Electronic Engineering as it requires a good knowledge of sampling theorem, filters and interpolation. MATLAB is central to this project thus improves my knowledge of a practical tool utilised in many electronic engineering disciplines. The third year communication systems course is also beneficial when evaluating the context of the DAC and the waveforms being generated. The testing phase of the GUI will involve using electronic lab equipment, such as, a signal generator and spectrum analyser which have been mentioned and used throughout the MEng course thus far and producing a user guide for the software will also utilise technical communication skills.

Content

The novel solution is to produce a MATLAB graphical user interface (GUI) to add clarity to the process. The GUI presents space to enter all of the required constants clearly with an explanation and units for each parameter. Some parameters, such as sine wave amplitude contains the option to choose which unit to input the parameter, in this case in LSBs (Least Significant Bits) or dBFS (Decibels Full Scale). There are options to create a frequency domain plot of the vector and/or save the vector to file which will generate an automatic, meaningful filename for the proposed vector. This is all carried out through the GUI thus eliminating the need to use MATLAB directly. The new scripts also include validation checks, for example if the sine wave frequency entered is greater than the Nyquist frequency, then a message box will provide a warning message.

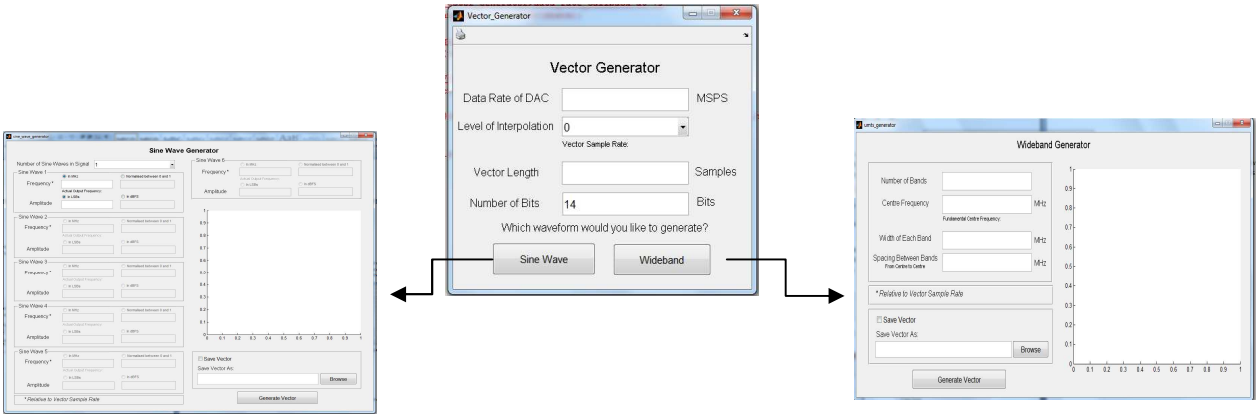

Figure 1. The main vector generator can be used to open the sine wave generator or the wideband generator

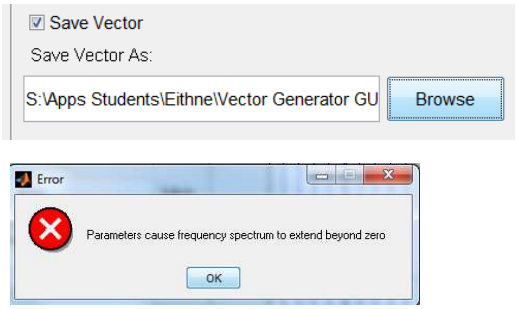

Figure 2. An automatically generated filename and an error message

The solution is to be tailored to the current DAC product; a 14 bit, 12GSPS DAC which contains two FIR filters which interpolate the input signal by 2 and can be set into high pass, low pass or all pass modes. Within the scope of this project, the GUI will be programmed to recommend which filters to use depending on the input information given. For example, if a user where to specify using 1 interpolation filter, with data rate of 12GSPS and a sine wave with frequency 7GHz then the software will be able to produce a message box recommending that the applied filter be used in high pass mode.

Figure 3.The interface can give messages to advise which filter mode the DAC should be set into.

The software will need to be tested by producing vectors, passing them through the DAC and checking the output on a spectrum analyser. There will also need to be a user manual for the software to ensure others can use the GUI after the six month placement.

Planning

This project has been completed and released for use within the department. It was completed in around 4 weeks. A future aim with this project may be to debug problems with the MATLAB code as they arise. Another responsibility has included producing vectors for customers on behalf of colleagues. This involves generating the vectors and checking in the lab that they are correct using a development board and spectrum analyser. This responsibility will be ongoing throughout the six month placement.

Development Board (1) ADC Replacement

Context

Each ADC or DAC, designed within the department, comes with a development board for customers. The development board has various purposes; it allows customers to check the performance of the component before integrating as part of a system and ensure it is fit for their purposes. These boards are designed and laid out in the applications department in Maidenhead. On each development board, there is a generic, externally designed ADC which is separate from the main ADC component designed by Fujitsu. The purpose of this generic ADC is to process signals on the board.

On development board (1) this ADC was a part known as MAX1034. This part was problematic as at low voltages it was forcing a higher voltage back onto the output pin. For example, when trying to apply a voltage of less than 0.2V to MAX1034 from a power supply, then the supply would still display 0.2V as something was forcing this particular voltage. After various troubleshooting measures from another engineer it was decided that the best way to proceed was to seek an alternative part. Therefore my role was to seek an alternative part and make the necessary changes to integrate this onto the development board.

This project is relevant to MEng electrical and electronic engineering as it requires finding a device to meet a relevant specification. It also involves designing a correct schematic and layout for the new part design and soldering the new parts onto a PCB. There is also a digital design element as the ADC control interface needs edited to reflect the new ADC device which involves interpreting timing diagrams then editing and simulating VHDL code. The project also requires editing the MATLAB script which interfaces between the development board and the computer.

Content

The first step in this project is to find a suitable replacement part. The given specification states that the device must be bipolar, have at least 12 bit resolution and have 8 channels. It must also be useable with an SPI interface. After comparing several suitable parts it was decided to use the AD7329.

The cheapest way to test if the new AD7329 corrects the problem given above is to create a separate milled board and use wires to connect pins on the milled board to test points on the main board. The milled board is designed using RS Design spark by doing a schematic of the recommended design from the AD7329 data sheet and a PCB layout. A gerber file is then produced and the PCB can be milled using a milling machine in the lab. The components needed can be ordered from Farnell website and soldered by hand onto the milled board.

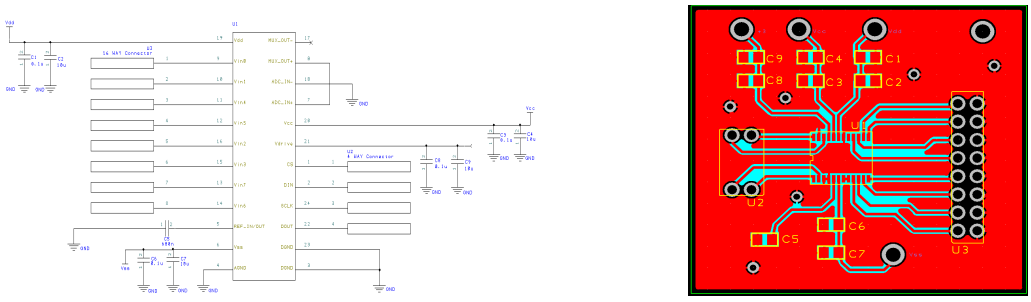

Figure 4. Recommended schematic and PCB layout for AD7329 designed using RS design spark

The control interface for the ADC comes from the field programmable graphics array (FPGA) on the main board. The control interface for the AD7329 is different from that of the MAX1034 and therefore certain alterations are needed. For example, AD7329 has 4 registers whereas MAX1034 has only 1 and the data out for AD7329 is 16 bits whereas for MAX1034 the output data is 32 bits so both the FPGA and MATLAB code need to be updated to reflect these changes.

The FPGA code is written in VHDL and can be edited using the Xilinx software, ISE. When using the AD7329, 3 registers must be set before we can trust the value at the output. There are 2 range registers where range register 1 is used to set the range of output values for channels 0 to 3 and range register 2 is used to set the range of output values for channels 4 to 7. For the purpose of this project we want to set all channels to ±5V mode.

We also need to set the control register this determines which channel we want to take the voltage for conversion from, whether we want differential or single ended mode and whether we use 2's complement. In our case, we want to use single ended and normal power mode we do want to use

2’s complement and an internal reference. The channel to read from is changeable and set in the MATLAB code.

To check that the new VHDL code works correctly a ModelSim simulation is beneficial. This involves first creating a ModelSim test bench and running a simulation. Analysis of the wave diagram identified a problem with the ADC CLK as it was idylling in the wrong state when chip select was high; this can be easily corrected in the VHDL code.

Finally the MATLAB code used to communicate between the development board and the computer needs to be updated to reflect the changes made above. A new variable must be added to the MATLAB SPIReadADC function to allow the user to select which register it wants to write to. The function already contains input variables to select which of the 2 AD7329’s on the board you wish to communicate with and which channel on that AD7329 you want to read from. The output of the function is the conversion result from the AD7329.

The way the MATLAB code processes resulting data also needs updated, the new code is shown below. If the data received is negative (signed_bit==16) then the resulting data should be calculated in one way and in a different way if the result is positive. This is because we are using 2’s complement numbers.

```
if signed_bit == 16
        result = ((((256*x)+DataIn(2))-4096)/4096)*5;
else
        result = (((256*x)+DataIn(2))/4096)*5;
end
```

Planning

The project has been completed as it has been shown that the AD7329 component can carry out a conversion of the input voltage right down to 0V i.e. it can convert values such as 0.03V correctly. The completion time for this project was about 5 weeks, largely due to inexperience with VHDL and the software used. A future action on this project will be over the coming weeks as a new revision of development board (1) will arrive containing 2 AD7329 laid out on the main board. These will need to be tested to ensure the component works correctly when integrated with the development board. This can be done using 3 power supplies, a USB cable and a computer with MATLAB therefore finding necessary equipment should not be problematic.

Development Board (2) SPI Interface

Context

Development board (1) from the previous section contains space to evaluate a Fujitsu ADC only. The next generation of this board, development board (2) allows space to evaluate both the ADC and corresponding DAC. This means that the functionality provided by the FPGA needs to be extended. This project will address the changes needed to update the interface. In scope, there will also be a new VHDL script in the FPGA project to control signals going to a microcontroller on the power supply board which is able to enable/disable supplies and change voltage values.

This project is again very relevant to MEng electrical and electronic engineering as it covers a range of areas, particularly digital system design. It provides more opportunity to practice VHDL and MATLAB programming as well as developing a deep understanding of the functionality of a development board.

Content

Development board (2) will have 4 of the generic ADCs (mentioned in the previous section of the report) whereas on development board (1) there are only 2 generic ADCs. There are more generic ADCs on the new board as there will be more signals to process. On development board (1), the ADC to read and write to, is chosen using a 2 bit chip select signal, CS_IN, where 1 bit will go high to select ADC(1) and the other will go high to select ADC(2). When the CS_INT signal goes low to indicate there is information available for the ADC, then the ADC_BI_CS output signal will be low and the selected chip will become active.

This approach is no longer feasible when there are 4 ADC's as the chip select comes into the FPGA on an 8 bit signal, where all 8 bits already carry data therefore the CS_IN signal must remain 2 bits wide. The solution is to use all 4 possible combinations that 2 bits can take as follows:

```
ADC_BI_CS(1) <= not(not(CS_IN(1)) and not(CS_IN(2)) and not(ADC_CS_INT));
ADC_BI_CS(2) <= not((CS_IN(1)) and not(CS_IN(2)) and not(ADC_CS_INT));
ADC_BI_CS(3) <= not(not(CS_IN(1)) and (CS_IN(2)) and not(ADC_CS_INT));
ADC_BI_CS(4) <= not((CS_IN(1)) and (CS_IN(2)) and not(ADC_CS_INT));
```

This logic has been added to the SPI interface and simulated in Modelsim to check that the result is correct.

As development board (2) will have both a Fujitsu developed ADC and DAC then 2 separate SPI interfaces must exist to communicate with each device separately. As the signals we are dealing with are at the output of the FPGA then the update must be in the top file of the ISE project and not in a

separate block. To complicate matters the SPI interface also has an external mode where the FPGA simply passes external signals through the FPGA without doing any processing.

This means that we want SPI_OUT signal to be mapped to the ADC_SPI_MISO pin on the FPGA, when the ADC ID is high and the external ID is low. Similarly if the DAC IC is high and the external ID is low then SPI out will be mapped to DAC_SPI_MOSI. If the external ID signal was to go high we would map the SPI_OUT to EXT_SPI_MOSI. A pin not being used will be set into high impedance mode. An example of this is shown below.

```
ADC_SPI_MOSI <= SPI_OUT when (EXT_SPI_CTRL_IN='0' and ID_ADC='1' and ID_DAC='0') else
EXT_SPI_MOSI when (EXT_SPI_CTRL_IN='1' and ID_ADC='1' and ID_DAC='0') else 'Z';

DAC_SPI_MOSI <= SPI_OUT when (EXT_SPI_CTRL_IN='0' and ID_ADC='0' and ID_DAC='1') else
EXT_SPI_MOSI when (EXT_SPI_CTRL_IN='1' and ID_ADC='0' and ID_DAC='1') else 'Z';
```

This is carried out for all input and output signals to the ADC and DAC. These need to be tested in Modelsim to ensure the outputs are correct.

An extension of this project has been to design a new SPI interface between the FPGA and a microcontroller on the power supply board to the main development board. The purpose of this interface is to allow control over different power supplies from MATLAB on the computer. It will allow supplies to be switched on and off remotely as well as changing the voltage values. This part of the project will involve creating a new control block, PSU_CONTROL in the current SPI Interface and editing the current main SPI_CONTROL block to take into account this new block. It will also involve writing a new MATLAB function to send the control data to the FPGA over USB.

Finally, this project will involve checking the names of all inputs and outputs on both versions of the FPGA schematic and if necessary, changing the labels in the VHDL code.

Planning

This project is ongoing and is expected to take around 5 weeks to complete. The code updates for 4 generic ADC's have already been completed and the changes to implement two separate SPI interfaces for the ADC and DAC have also been finished and simulated in Modelsim. Currently, I am working on the SPI interface between the FPGA and microcontroller on the power supply board. The PSU_CONTROL script has been written in VHDL using ISE software and now needs to be simulated in Modelsim. The MATLAB function to allow the power supply information to be sent to the FPGA still needs to be written. No hardware testing of the project has been completed yet.

It is likely that I will need ongoing support from my supervisor to complete this project. As he has written the original SPI interface, he is able to answer any questions. This will most likely take the form of informal communication when it is needed rather than scheduled meetings. The interface will need tested when the new development board (2) arrives; as this arrival is scheduled for October, my placement will already have finished therefore this will be a role for another member of the applications team.

Conclusion

Overall this placement thus far has involved several worthwhile and interesting projects which are very relevant to MEng Electrical and Electronic Engineering. I have particularly enjoyed learning VHDL and applying this to the problems that to the digital problems I have been given. I think the placement has been very beneficial in giving an insight into the electronics industry and I feel that it has improved my overall electronics knowledge and confidence as a professional engineer.

In the remaining two months, apart from finishing the current project, there is no set work plan for this time. Normally, when I finish a project, my supervisor and I will have a discussion about available work and what area I would like more experience in. This allows both parties to get the most out of the placement and formulate a plan for the following number of weeks. I will finish my current project within two weeks and gain a better idea of a new project then.