TÉCNICO LISBOA

Projecto e Controlo em Lógica Digital

5º Laboratório

Microprocessador

O objectivo deste trabalho consiste na implementação de um microprocessador dentro da FPGA da placa DE2 com vista a realizar o interface de um osciloscópio. Neste trabalho pretende-se implementar um processador NIOS II que corre um pequeno programa de exemplo. Este programa deve ler os estados dos interruptores e activar os leds de acordo com o estado dos interruptores. Além disso o estado dos interruptores deve ser mostrado no computador que está ligado à placa DE2.

Genericamente os passos são:

- Implementar um processador (e demais hardware) usando a ferramenta QSYS do Quartus II.
- Programar o microprocessador na placa DE2
- Fazer um projecto de software usando o "NIOS II Software Build Tools for Eclipse"
- Escrever um pequeno programa em C que realize o pretendido

A utilização de um microprocessador numa FPGA consume recursos e, sendo um sistema sequencial, não o desempenho que tem um sistema implementado em Verilog. No entanto facilita bastante o uso de periféricos que necessitam intrinsecamente de ser controlados sequencialmente.

Esta implementação é algo morosa sendo necessário um conjunto extenso de configurações de parâmetros. Para minimizar este tempo é dada uma receita, baseada em tutoriais da Altera. A receita está testada para a placa DE2 utilizando o Quartus II e o Eclipse na versão 12.

Os tutoriais utilizados são:

- "Introduction to the Altera Qsys System Integration Tool"
  Altera corporation
  ftp://ftp.altera.com/up/pub/Altera_Material/12.0/Tutorials/Introduction_to_the_Altera_Qsys_Tool.pdf
- " Using the SDRAM Memory on Altera's DE2 Board with Verilog Design"
  Altera Corporation
  ftp://ftp.altera.com/up/pub/Tutorials/DE2/Computer_Organization/tut_DE2_sdram_verilog.pdf
- "My First Nios II Software Tutorial"
  Altera corporation
  http://www.altera.com/literature/tt/tt_my_first_nios_sw.pdf

Tutorial para a implementação do sistema

Início do projecto

Comece com o projecto DE2_TOP "limpo". Atenção que algumas ferramentas não admitem espaços nos caminhos das directorias. Sugestão: crie no **c:** uma pasta grupo_letra e desenvolva aí o projecto.

Abra o projecto no QuartusII.

Definir o sistema do NIOS II usando a ferramenta Qsys

Lance a ferramenta Qsys a partir do menu "tools". Deverá aparecer um novo projecto apenas com um clk.

Ir ao separador "Clock Settings" e confirmar que existe um relógio; external; 50 MHz.

Voltar ao separador "System Contents".

Especificar o Microprocessador

No "Component Library" expandir "Embedded Processors"

Seleccionar "NiosII processor"; clicar em "Add"[1]

Escolher "Nios II/s"

Em "Hardware Arithmetic Operation" escolher "none" para "Hardware Multiplication Type" e não seleccionar "Hardware Divide".

Clicar em "Finish" para voltar à janela principal.

Implementar o módulo SDRAM

Adicionar: "Memories and Memories controllers"→"External Memory Interfaces"→"SDRAM interfaces"→"SDRAM Controller"

Na Janela de configuração escolher:

- "Presets": "Custom"
- "Data Width": 16 bits
- Deixar o resto com os valores pré-definidos.
- Clicar em "Finish"

Implementar Portos paralelos

Adicionar: "Peripherals"→ "Microcontroller Peripherals" → "PIO (Parallel I/O)"

Na Janela de configuração escolher:

- "Width": 8
- "Direction": "input"
- Clicar em "Finish"

Como é um porto de entrada, alterar o nome na janela principal para "switches".

Adicionar: "Peripherals"→ "Microcontroller Peripherals" → "PIO (Parallel I/O)"

Na Janela de configuração escolher:

- "Width": 8
- "Direction": "output"
- Clicar em "Finish"

Como é um porto de saída, alterar o nome na janela principal para "Leds".

Implementar JTAG UART

Trata-se de um interface série que vai permitir a comunicação com o computador pelo cabo de programação da DE2. Poderíamos implementar outros protocolos, e.g. RS232.

[1] Na janela de configuração devem aparecer erros pois ainda não existe a memória sobre a qual o microprocessador vai correr. Será corrigido mais à frente.

Adicionar: “Interface Protocols” → “Serial” → ”JTAG UART”

Não alterar nenhuma configuração e clicar em “Finish”.

Ligações

Na janela principal temos agora os componentes todos mas falta implementar as ligações entre os diferentes componentes. Tipicamente o clock de todos os componentes provém do “clock ouput” do clk_0; o reset vem do “reset ouput” do clk_0 e do “reset ouput” do “jtag debug module” (está dentro do componente do nios II; não confundir com o JTAG UART).

- Ligar o “clock ouput” do clk_0 a:
  - “clk” do nios II
  - “clk” da sdram
  - “clk” dos leds
  - “clk” dos switches
  - “clk” do jtag uart
- Ligar o “reset ouput” do clk_0 e do “reset ouput” do “jtag debug module” a:
  - “reset” do nios II
  - “reset” da sdram
  - “reset” dos leds
  - “reset” dos switches
  - “reset” do jtag uart
- Ligar o “s1” da memória ao “data master” e “instruction master” do niosII
- Ligar o “s1” dos leds e switches ao “data master” do nios II
- Ligar o “avalon jtag slave” do “JTAG UART” ao “data master do nios II
- Ligar a linha de IRQ do JTAG UART ao Nios II (seleccionar na coluna IRQ) usando o porto 5

Definir endereços

Uma das colunas do Qsys é chamada de “base”. Aqui são apresentados os valores base das regiões de memória. Podem ser configuradas manualmente ou pelo sistema.

Configurar a memória SDRAM para ter o endereço base 0x00000000.

Bloquear este endereço clicando no cadeado

Ir a “System” e escolher “Assign Base Adresses”

Tome nota dos valores atribuídos. Em particular para os leds e para os switches

Configurar a memória do Microprocessador

O processador Nios II usa dois vectores para saber onde estão as instruções a executar. São eles o “reset vector” e o “exception vector”. São implementados na SDRAM. Agora que os endereços de memória estão atribuídos podemos configurar o Nios II.

- Editar o processador Nios II
- Em “Reset Vector” escolher
  - “Reset Vector memory”: sdram_0.s1
  - Não alterar os valores default do offset nem do reset vector (deverá corresponder ao endereço base da memória)
- Em “Exception Vector escolher
  - “Exception Vector Memory”: sdram_0.s1
  - Não alterar os valores default do offset nem do reset vector (deverá corresponder ao endereço base da memória somado do offset)[2]
- Clicar em “finish”

Exportar portos para a FPGA (para o DE2_TOP)

Vamos precisar de exportar para o projecto os portos correspondentes aos switches, aos leds e à SDRAM. Na coluna “export” aparece a cinza “click to export”…

- Clicar em “click to export” na linha “external connection” do porto dos “switches” e dar um nome, p.ex. “switches”.

[2] Após esta configuração deverão desaparecer os erros de memória.

- Clicar em “click to export” na linha “external connection” do porto dos “leds” e dar um nome, p.ex. “leds”.
- Clicar em “click to export” na linha “wire” do porto da “sdram_0” e dar um nome. p.ex. “sdram”.

Os nomes dados vão aparecer como portos do módulo verilog gerado.

Gerar o módulo verilog

Guardar o sistema. Certifique-se que guarda na directoria correcta.

Em seguida ir ao separador “generation”. Seleccionar “none” em “Create Simulation Model” e em “Create testbench Qsys system”

Clicar em “Generate”[3]

Sair do Qsys e voltar ao Quartus II

Integrar o sistema no DE2_Top

Gerar os clocks necessários para a SDRAM

A defasagem do clock depende das características físicas da placa DE2. Para o funcionamento adequado do chip SDRAM é necessário que o seu clock, DRAM_CLK, preceda o relógio do sistema Nios II, CLOCK_50, por 3 nanosegundos. Isto pode ser conseguido através do uso de um PLL. Use o “Mega Wizard” para criar um PLL que tem como input o clock de 50MHz e como output dois clocks com a mesma frequência: o primeiro adiantado de 3ns (vai ser o clock da SDRAM) e o segundo em fase com a entrada (Clk do niosII).

Instanciar o módulo gerado no Qsys

No Quartus II abrir o ficheiro verilog gerado. Nesta fase o ficheiro Não deve estar incluído no projecto. Abrir da directoria. O ficheiro deve estar dentro de uma subdirectoria do DE2_TOP. P.ex. nios_system/synthesis/nios_system.v

Identificar a definição do módulo e os portos utilizados. As ligações necessárias devem ser obvias à excepção de dois detalhes nos sinais da SDRAM:

- “sdram_ba” é um vector de 2 bits no sistema nios enquanto que no DE2_TOP são dois sinais de 1 bit (DRAM_BA_1 e DRAM_BA_0). Concatene-os.
- “sdram_dqm” é um vector de 2 bits no sistema nios enquanto que no DE2_TOP são dois sinais de 1 bit (DRAM_UDQM e DRAM_LDQM). Concatene-os.

Não esquecer de ligar o clock da SDRAM e do Nios vindos do PLL. Não esquecer o Reset e os portos dos leds e dos switches. Retirar o assignment de DRAM_DQ a alta impedância.

Compilar o projecto. (pode ser necessário adicionar o ficheiro do Qsys ao projecto. Neste caso nios_system.qsys)

Programar o projecto. O Quartus deve dar um aviso que tem um core com uma licença limitada. Aceitar e deixar a janela aberta para manter o programa. Não sair do Quartus.

Aplicação de software

Abrir o NIOS Software Build Tools for Eclipse

Criar uma directoria para o Workspace do Eclipse debaixo da directoria do projecto.

[3] Se for necessário é possível reconfigurar o sistema reabrindo o Qsys.

Criar o "Board Support Package"

O board Support Package é a parte do software que vai conter a informação sobre o sistema NIOS implementado na FPGA.

File→New→Nios II Board Support Package

- Dar um nome ao projecto. E.g. soft_bsp
- Em "SOPC Information file name" escolher o ficheiro ".sopcinfo" gerado. Neste caso "nios_system.sopcinfo". Depois de lida a file deverá aparecer em CPU o cpu implementado (nios2_qsys_o neste caso)
- Deixar os defaults e clicar em finish

Criar a aplicação

File→New→Nios II application

- Dar um nome ao projecto. E.g. soft
- Em BSP location procurar pelo bsp criado (soft_bsp)
- Deixar os defaults e clicar em finish

No project explorer clicar com o botão direito na aplicação (soft) e fazer New→Source File.
Dar um nome e.g. app_main.c
Clicar em finish.

Criar o código c

Os Leds e os Switches comportam-se como espaços de memória que podem ser escritos e/ou lidos. No nosso caso o endereço de memória é definido pelo Qsys (p.ex. na compilação de teste temos leds em 0x00801000 e os switches em 0x00801010. Cada um com 8 bits o que corresponde a um char.

Comecemos por definir os leds e os switches:

```
#define switches (volatile char*) 0x00801010
#define leds (char*) 0x00801000
```

No ficheiro app_main.c criar a função main que será executada pelo processador:

```
int main (int argc, char** argc1, char** argv2)
```

Seguidamente criar um loop contínuo, p.ex. while(1) em que se copia o conteúdo apontado pela variável switches para o espaço de memória apontado pela variável leds:

```
*leds=*switches;
```

Gravar o ficheiro![4]

Compilar e correr

Clique com o botão direito em cima da aplicação (soft) e fazer "build project"

Clique com o botão direito em cima da aplicação (soft) e fazer:
"Run As → Nios II Hardware"

Ao reconfigurar os switches os leds devem ser acendidos e apagados. Pode agora tentar um getchar e printf dentro do while. A consola do nios comporta-se como stdout e stdin…

[4] Atenção que o eclipse pode não avisar ser necessário gravar o ficheiro, compilando sem gravar as alterações