Laboratório 5

Usando a UART e Temporizadores

Neste laboratório veremos como enviar e receber dados de dispositivos de entrada e saída através do circuito UART, e como gerar eventos através de temporizadores. Usaremos o sistema *DE2 Basic Computer* nos experimentos, que já conta com os circuitos necessários de UART e temporização.

Introdução

A transferência serial de dados entre um processador e um dispositivo de entrada e saída é comumenente realizada através de um circuito conhecido como UART (*Universal Asynchronous Receiver Transmitter*). Uma interface UART é colocada entre o processador e o dispositivo de entrada e saída. Nesse esquema um dado de 8 bits (caracter) é manuseado por vez. A transferência do dado entre a UART e o processador é feita em paralelo, onde todos os bits do caracter são transferidos ao mesmo tempo usando diferentes fios. No entanto, a transferência do dado entre a UART e o dispositivo de entrada e saída é feita de forma serial, transferindo um bit por vez.

O sistema *DE2 Basic Computer* disponibiliza uma interface do tipo UART, conhecida como JTAG UART, para estabelecer uma conexão entre o processador Nios II e o computador hospedeiro conectado à placa DE2. A Figura 1 mostra o diagrama de blocos do circuito com a JTAG UART. De um lado a JTAG UART conecta-se com o barramento Avalon, que interconecta o processador Nios II, os circuitos de memória e as interfaces de entrada e saída. De outro lado ela conecta-se ao computador hospedeiro através da interface USB-Blaster.

O núcleo da JTAG UART contém dois registradores: *Data* e *Control*, que são acessados pelo processador pelo esquema de memória mapeada. O endereço do registrador *Control* é 4 bytes maior do que o endereço atribuído ao registrador *Data*. O núcleo da UART também contém duas filas (FIFOs) que servem como buffers, uma para armazenar os dados sendo transmitidos para o hospedeiro, e outra para armazenar os dados recebidos do hospedeiro. A Figura 2 mostra o formato desses registradores.

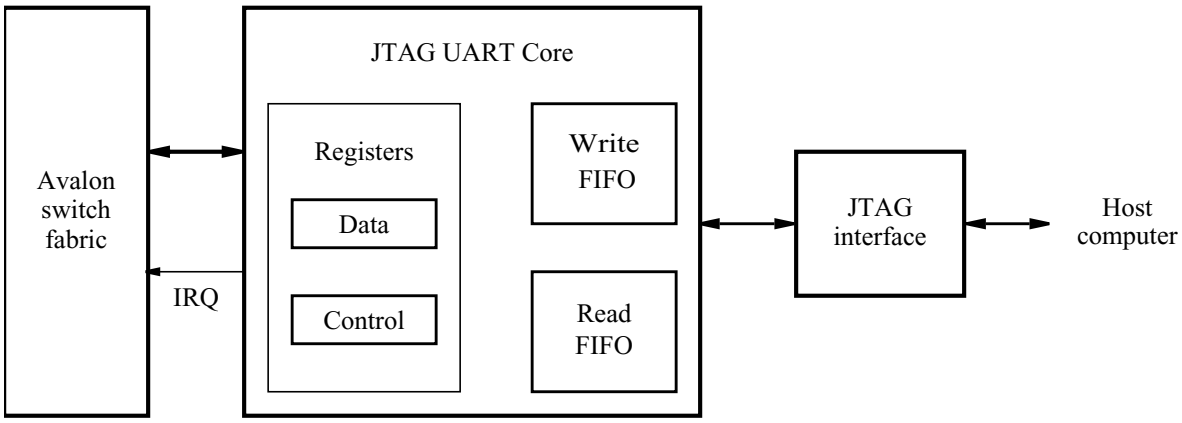

Figura 1. Diagrama de bloco do circuito da JTAG UART

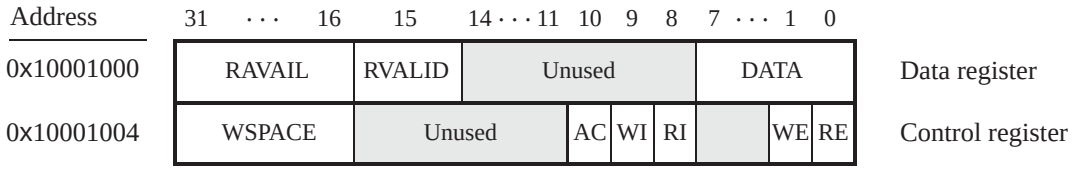

Figura 2. Registradores da JTAG UART

Os campos do registrador *Data* são:

- b_{7-0} (DATA) é um caracter de 8 bits a ser colocado na FIFO de escrita quando uma operação de armazenamento é realizado pelo processador, ou é um caracter lido da FIFO de leitura quando a operação de carga (load) é realizada.
- b_{15} (RVALID) indica se o campo DATA contém um caracter válido que pode ser lido pelo processador. Este bit é configurado para 1 se o campo DATA é válido; do contrário ele contém o valor 0.
- b_{31-16} (RAVAIL) indica o número de caracteres restantes na FIFO de leitura (depois desta leitura).

Os campos do registrador *Control* são:

- b_0 (RE) habilita interrupções de leitura quando 1.
- b_1 (WE) habilita interrupções de escrita quando 1.
- b_8 (RI) indica que uma interrupção de leitura está pendente quando o valor for 1. A operação de leitura do registrador *Data* limpa este bit (torna-o 0).
- b_9 (WI) indica que uma interrupção de escrita está pendente se o valor for 1.
- b_{10} (AC) indica que houve atividade da JTAG (como, por exemplo, o computador hospedeiro checando (polling) a JTAG UART para verificar que uma conexão existe) desde que o bit foi limpo. Escrevendo 1 em AC torna-o 0.
- b_{31-16} (WSPACE) indica o número de espaços disponíveis na FIFO de escrita.

Mais informações sobre a JTAG UART podem ser encontradas no Capítulo 5 da referência *Altera Embedded Peripherals Handbook*, disponível no site da disciplina.

Neste laboratório vamos usar a JTAG UART para transferir caracteres ASCII entre o processador Nios II implementado na placa DE2 e o computador hospedeiro. Também usaremos um temporizador para fornecer atrasos regulares. Um diagrama de blocos do hardware que queremos é mostrado na Figura 3. Este hardware está presente no *DE2 Basic Computer*.

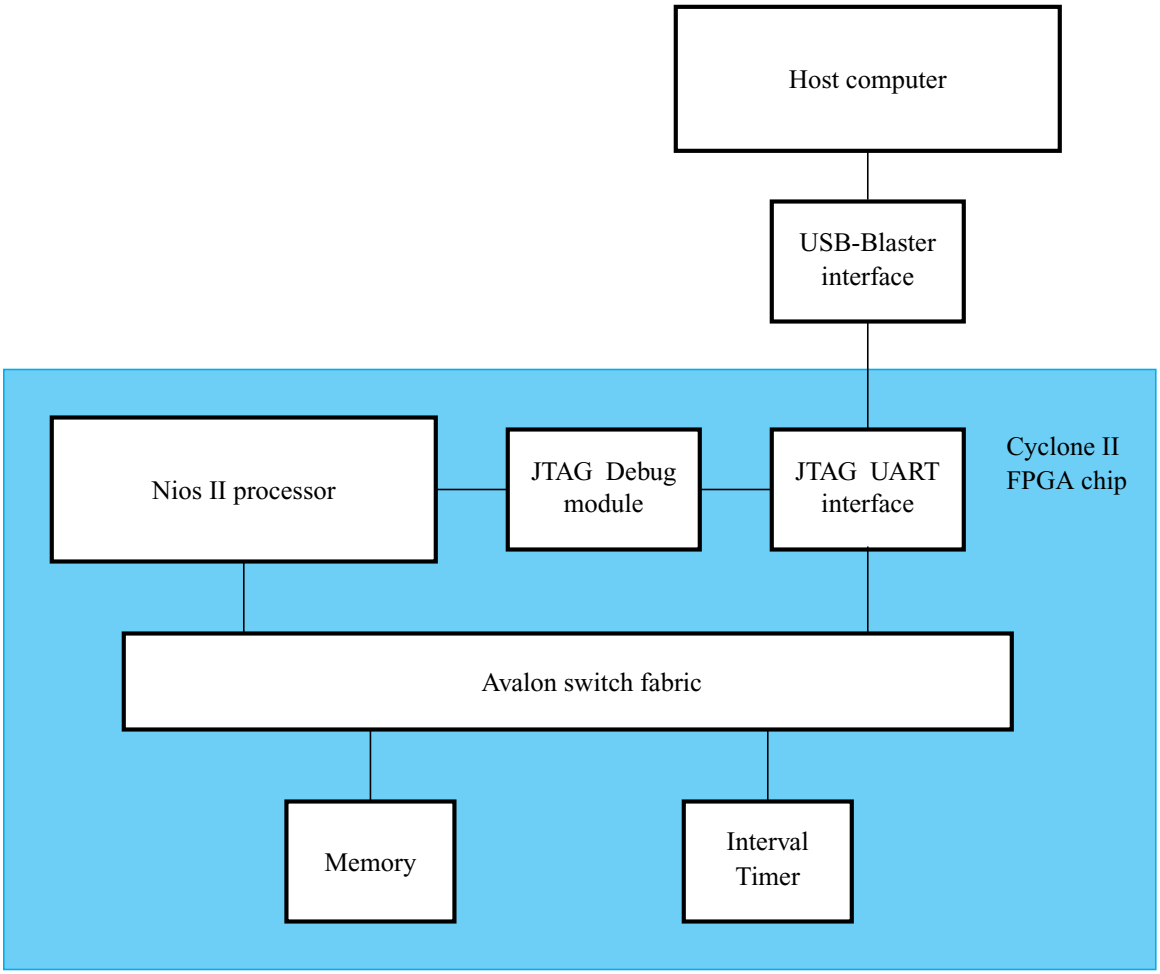

Figura 3. O sistema Nios II desejado mostrado em um chip de FPGA Cyclone II

Parte I

O componente JTAG UART pode enviar caracteres ASCII para o programa Altera Monitor, o qual imprimirá esses caracteres na janela do terminal. Quando o campo WSPACE do registrador *Control* da JTAG UART possuir um valor diferente de zero, a JTAG UART pode aceitar um novo caracter a ser escrito no programa Altera Monitor. Para escrever um caracter no Altera Monitor, use a técnica de polling para continuamente ler este registrador (*Control*) até que haja espaço disponível. Havendo espaço disponível, o caracter ASCII pode ser escrito no registrador *Data* da JTAG UART.

Escreva um programa em linguagem de montagem do Nios II para mostrar a letra Z aproximadamente a cada meio segundo na janela do terminal do programa monitor. Crie e execute o programa como a seguir:

1. Usando a linguagem de montagem do Nios II, escreva um laço que leia o registrador *Control* da JTAG UART e permaneça no laço até que haja espaço para escrita disponível.
2. Escreva a letra Z no registrador *Data*.
3. Usando o programa monitor, compile e carregue o programa na placa DE2.
4. Execute o programa em modo passo-a-passo. Se você executar esse programa usando o modo Continue, o caracter será enviado à janela do terminal mais rápido do que o programa monitor consegue manuseá-lo.
5. No código em linguagem de montagem, crie um laço de atraso de modo que caracteres sejam impressos somente em aproximadamente meio segundo.

6. Recompile, carregue e execute o programa. Discuta no seu relatório as questões envolvidas na geração do laço de atraso.

Parte II

A JTAG UART pode tanto receber caracteres ASCII da janela do terminal como enviá-los para impressão. O bit RVALID, b_{15}, do registrador *Data* indica se um valor no campo DATA é um caracter ASCII recebido válido. Se mais caracteres ainda estão esperando para serem lidos, o campo RAVAIL terá um valor diferente de zero.

Escreva um programa que implementa uma tarefa do estilo "máquina de escrever"; ou seja, leia cada caracter que é recebido pela JTAG UART a partir do computador hospedeiro e então mostre o caracter na janela do terminal do programa monitor. Use a técnica de polling para determinar se um novo caracter está disponível na JTAG UART.

Nota: o cursor deve estar na janela do terminal do programa monitor para que seja possível escrever caracteres na porta de escrita da JTAG UART.

Parte III

Nesta parte desejamos escrever um programa para ler os caracteres recebidos pela JTAG UART do computador hospedeiro, e imprimir o último caracter recebido repetidamente a cada 500 milisegundos. Na Parte I nós usamos um laço de atraso para gerar um intervalo de tempo aproximado com essa duração. Agora, vamos usar um circuito temporizador para esse propósito. O temporizador deve interromper o processador a cada 500ms, instante no qual um caracter deve ser escrito na janela do terminal do programa monitor.

O temporizador possui um contador interno no qual um valor específico é configurado e então decrementado a cada ciclo do relógio. Quando o contador atinge o valor 0, um evento de "timeout" é dito ocorrer. Neste ponto, o temporizador pode gerar um pedido de interrupção e o contador pode ser resetado para o valor especificado. O temporizador possui um conjunto de registradores de 16 bits que podem ser acessados como se fossem posições de memória, similar ao da JTAG UART. Estes registradores são mostrados na Figura 4. O endereço do registrador de *Status* é 0x10002000, que é o endereço base atribuído ao temporizador. O registrador *Control* está no endereço 0x10002004. O valor inicial do contador é especificado nos registradores presentes nos endereços 0x10002008 (16 bits menos significativos) e 0x1000200C (16 bits mais significativos).

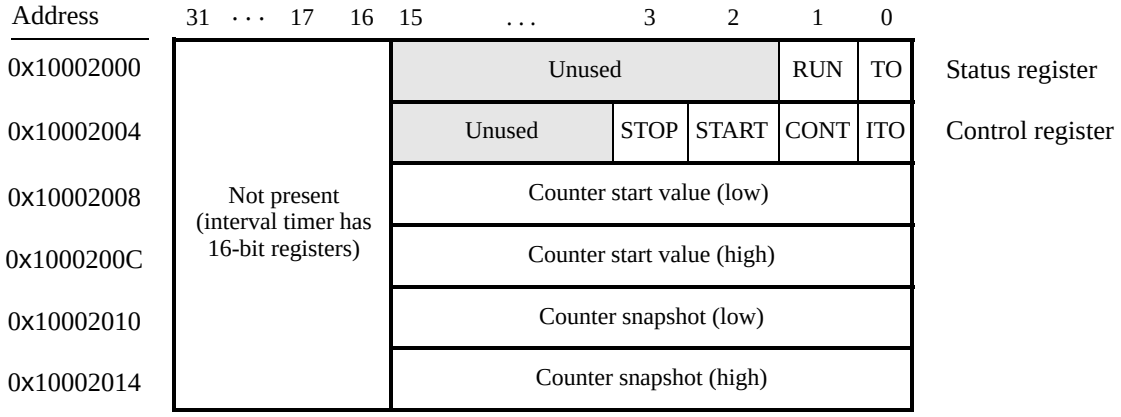

Figura 4. Registradores do Temporizador

Os bits do registrador *Status* são usados da seguinte forma:

- b_0 (TO) é o bit de timeout. Ele fica em 1 quando o contador interno do temporizador atinge o valor 0. Ele permanece em 1 até que seja explicitamente limpado pelo processador (escrevendo o valor 0), o que deve ser feito para limpar uma requisição de interrupção existente.
- b_1 (RUN) é igual a 1 quando o contador interno está correndo; do contrário, é igual a 0. Este bit não é alterado pela operação de escrita no registrador *Status*.

Os bits do registrador *Control* são usados assim:

- b_0 (ITO) habilita interrupções do temporizador quando em 1.
- b_1 (CONT) determina como o contador interno se comporta quando atinge o valor 0. Caso CONT = 1, o contador funciona continuamente ao recarregar o valor de contagem especificado inicialmente; do contrário, ele para quando atinge 0.
- b_2 (START) faz com que o contador interno comece a contar quando configurado em 1.
- b_3 (STOP) para o contador interno quando configurado em 1.

Mais informações sobre o temporizador podem ser encontradas no capítulo 12 do *Altera Embedded Peripherals Handbook*.

Para habilitar ambas interrupções, do temporizador e da JTAG UART para leitura de caracteres (Parte III), os bits b_8 e b_0 do registrador de controle *ctl3* devem ambos estar em 1. O registrador de controle *ctl4*, também conhecido como *ipending*, pode ser usado para determinar qual interrupção ocorreu. Se uma interrupção for desabilitada através do registrador de controle *ctl3*, a rotina de tratamento de interrupção não será executada e também não será mostrada como sendo disparado no registrador de controle *ctl4*, nem mesmo se o dispositivo estiver requisitando interrupção.

Prossiga como a seguir:

1. Crie uma rotina de tratamento de interrupção para lidar tanto com a interrupção do temporizador quanto da interrupção de leitura da JTAG UART.
2. Para habilitar interrupções, valores apropriados devem ser escritos no registrador *Control* da JTAG UART, no registrador *Control* do temporizador e nos registradores de controle *ctl0* e *ctl3* do Nios II.
3. Compile, carregue e execute seu programa.
4. Discuta no seu relatório as vantagens e desvantagens em se usar um temporizador ao invés de um laço de atraso.