UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO" - UNESP
BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO
Microprocessadores

Marina Barbosa Américo
João Pedro Brum Terra
João Victor Figueredo

Relatório trabalho do trabalho final de Microprocessadores

1. Introdução

Neste trabalho, diversos conceitos vistos durante a disciplina de microprocessadores foram utilizados para implementar as soluções desejadas do sistema. Para isso utilizamos o UART para controlar os comandos que são dados ao aplicativo, além de realizar o tratamento de interrupção, sistema de polling para administrar a comunicação UART quando recebe uma entrada no console e o cronômetro para a realização de animação de leds e acionamento da contagem de tempo.

2. Funcionalidades

Aqui está listado as funcionalidades implementadas.

2.1. Menu

Utiliza o console como forma de receber os inputs, tendo a mensagem “Entre com o comando:” sendo exibida no prompt, assim que receber os códigos relevantes, a UART realiza a comunicação do computador até a placa. Num sistema de polling ocorre a espera os comandos para acionar as sub rotinas desejadas.

2.2. (00) Acender o led

Ao receber o comando no console de **00xx**, sendo **xx** o enésimo led vermelho, o enésimo led é aceso.

2.2.1. (01) Apagar o led

Ao receber o comando no console de **01xx**, sendo **xx** o enésimo led vermelho, o enésimo led é apagado.

2.3. (10) Animação com leds

Ao receber o comando no console de **10** e a depender da chave **SW0** ele realizará uma animação de deslocamento. A direção da animação é definida pela chave **SW0**, se estiver em alto ele vai da direita para esquerda, se estiver em baixa vai da direita para a esquerda.

2.3.1. (11) Parar animação com leds

Ao receber o comando no console de **11** ele encerra a animação dos leds.

2.4. (20) Inicia o cronômetro

Ao receber o comando no console de **20**, ele inicia uma contagem decimal progressiva do cronômetro, mostrando em 4 displays os valores em segundos até 9999 segundos.

Enquanto o cronômetro estiver acionado, se o botão **KEY1** for pressionado ele vai pausar a contagem, retomando a contagem do valor em que estava anteriormente.

2.4.1. (21) Parar o cronômetro

Ao receber o comando no console de **21**, a contagem e o cronômetro são encerrados.

3. Implementação

Aqui está documentado o processo e as abordagens feitas para implementar cada uma das funcionalidades. Implementamos o projeto de forma que cada funcionalidade está desenvolvida em seu próprio documento à parte do menu principal, salvo o processo de animação dos LEDs, que existe junto do menu mas é configurado à parte em sua própria seção.

3.1. Inicialização

O programa inicializa carregando para os registradores os endereços do ponteiro da pilha, da JTAG UART, e da frase inicial.

Em seguida ele realiza a primeira leitura do JTAG UART e caso o byte não seja nulo ele realiza a escrita e permanece no loop, passando para o próximo byte.

Ao fim, quando a leitura dos caracteres acaba, ele vai para a região do GET_JTAG, que vai servir como menu em nossa implementação.

3.2. Menu

O menu se encontra no GET_JTAG. Lá é realizado um polling que fica esperando a leitura de um byte, realizando o load do “ldwio” e fazendo um andi para ver se há algum dado entrando para depois verificar no “beq” se é igual a zero, para que assim possa voltar para o topo do polling.

Caso haja um byte entrando ele passa o “beq”, extrai o byte menos significativo e inicializa a sub rotina PUT_JTAG que escreve um caractere na UART.

E por fim, é realizada uma comparação, onde o valor de cada opção é oferecido e é realizada uma comparação para verificar qual sub-rotina deverá ser chamada. Todos os valores são carregados via “movi” para o registrador r10 e é realizado diversas comparações até ele encontrar um valor igual ou voltar para topo do polling através de um branch (“br”).

3.3. PUT_JTAG

Essa sub-rotina é utilizada para escrever os bytes que foram recebidos na UART.
Ele começa reservando espaço na pilha e carregando a informação no registrador r4. Em seguida realiza a leitura no endereço no UART para depois verificar usando “andhi” se a parte alta do byte é igual a 0xffff e compara com zero. Se for igual a zero ele ignora e vai para fim da sub-rotina, deslocando o conteúdo de dentro do registrador r4, e libera espaço na pilha. Se houver, ele vai e realiza a escrita utilizando “stwio” e depois realiza a liberação de espaço.

3.4. Acender o led

Nesta sub rotina é implementado a funcionalidade de acender e apagar os leds.
Começa-se realizando um polling para aguardar a leitura do próximo byte. Recebendo o próximo dado ele extrai o byte menos significativo e o escreve na sub rotina PUT_JTAG.

A Partir daí, há uma comparação semelhante ao do menu carregando o segundo dígito para realizar a comparação e executar a funcionalidade pedida.

Quando aceso, pode-se acender múltiplos leds, assim como apagá-los.

3.5. Apagar o led

Quando chamado, essa sessão vai realizar o procedimento de esperar pelos dois dígitos finais e escrevê-los de volta na UART.

Em seguida é realiza a conversão dos dois últimos dígitos para obter o enésimo led que deseja acender, realizando em seguida a leitura do endereço de configuração do led e então se realiza o carregamento do valor correspondente a o enésimo led para ser carregado e assim acender o led na placa.

Assim como na funcionalidade de acender os leds, quando acionada, ela pode ser executada em conjunto com outros leds acesos

3.6. Animação com leds

Ele inicia na seção FIRST_FUNCTION, que esta no arquivo do menu após sair do GET_JTAG e é feito o polling para verificar se a animação deve ser ou não executada. Se sim, é então dado uma branch para uma sub-rotina chamada LEDS-ANIMATION

Nesta sub-rotina é feita a inicialização do timer, carregando para a memória os devidos endereços. Além disso, é feito a configuração das interrupções que serão geradas pelo timer para dar o sinal para o deslocamento do led

Em seguida, voltando para a interrupção definida no arquivo do menu é realizado o seu tratamento e verificado se ela foi provocada pela animação ou pelo cronômetro. Se for da animação então é realizada a leitura do primeiro switch para que assim define se a animação será para a esquerda ou direita. Caso o for ele desloca os bits conforme foram feitas as interrupções.

Já a sua pausa, o fim da animação, quando acionada vai para a sub-rotina STOP_ANIMATION, que desabilita o timer e envia uma nova linha para o terminal e assim, retornando para o laço do GET_JTAG.

3.7. Cronômetro

Aqui, a funcionalidade primeiro carrega, na seção do SECOND_FUNCTION, os caracteres para verificar se deve ser acionado ou parado o cronômetro, caso receba um zero ele vai para a sub-rotina LEDS_TIMER. É iniciado o timer, escrevendo no

endereço de configuração do cronômetro e da interrupção dele e o deixando disponível. Além disso, também é configurado o botão de pausa e retomada do timer.

No arquivo principal é feito uma comparação para determinar o uso do timer, e caso for para cronômetro, ele inicia a sub-rotina de CRONOMETRO_CONFIG.

Ao entrar na seção do CRONOMETRO_CONFIG, ele carrega para o registrador r15 o endereço do controle e realiza um stwio e realiza um incremento, para depois realizar uma comparação, caso a contagem atinge o valor 10, para poder dar branch para ADD_DEZENA.

Nesta seção realiza-se um incremento, para verificar se o valor atingido é igual a cem para então dar outro branch para ADD_CENTENA. Nesta seção é realizado um novo incremento e uma nova comparação que, se atingida, vai para ADD_MILHAR e por fim, nesta seção é feito uma comparação final para verificar se se o incremento atingiu o valor necessário para resetar a contagem na branch RESET_COUNTER.

Após a execução destas seções é passado para SHOW_COUNTER e carregado o array do display para o registrador r13. Depois é realizado um andi com o valor da máscara para o sete segmentos e em seguida é realizada a leitura do endereço dos segmentos corretos e depois realizada a escrita no endereço.

Esse processo é repetido para preencher todos a unidade, dezena, centena e milhar e após a escrita no display é dado o comando ret.

3.8. Pausa e retoma o cronômetro

Aqui a execução é feita com base no tratamento de interrupção em que se é lido o edge do botão e realizado a ação conforme o estado ali presente.

4. Considerações finais

Boa parte das funcionalidades que foram implementadas tiveram apoio no conteúdo visto em aula e também no guia do Altera DE2, lá foi possível se aprofundar na logica por tras dos métodos utilizados e entender melhor o processo das comunicações entre os dispositivos de entrada e saída.

A lógica e baseia em pollings que aqui agem como uma espécie de buffer, que fica aguardando o recebimento de instruções JTAG UART, onde nós temos um componente que se comunica externamente rápido e precisa se comunicar com um lento.

Em seguida foi utilizado diversas instruções de leitura e escrita de endereço que estão relacionados com a saída e a entrada.

Para o temporizador, foi utilizado o sistema de interrupção para realizar os comportamentos desejados pelo projeto. Lá se é controlado também a animação dos leds e o cronômetro,

havendo uma comparação para verificar qual das duas funcionalidades está operando no momento.
Vale ressaltar alguns comportamentos do sistema. A primeira e de que quando o cronômetro é iniciado, as demais funções não são mais ativadas e se há o congelamento delas até o cronômetro ser desligado. E também, quando iniciado a animação do led, o leds que já estavam acessos são desligados.

5. Conclusão

Com o desenvolvimento do projeto foi possível destacar que as funcionalidades mais complexas foram construídas a partir de métodos mais simples e primitivos, elaborando tudo que foi estudado em sala de aula, estudados nos laboratórios e atividades e vistos no material de apoio.

Além disso, também foi possível vislumbrar conceitos antes visto em linguagens de mais alto nível sendo aplicados em uma linguagem de baixo nível como estruturas de dados, manipulação de array e utilização de entradas e saídas.