

RELATÓRIO DO PROJETO FINAL

I) RESUMO

Nesse projeto foi desenvolvido uma espécie de aplicativo console que aceita comandos do usuário através da JTAG UART e, dependendo do comando inserido, uma certa ação é executada pelo código.

Para tornar o aplicativo mais iterativo, foi implementado um contador utilizando o display de 7 segmentos(onde eram colocados os dígitos) e o temporizador(que fazia as interrupções para compor a lógica de um relógio que conta até 60 minutos), também era colocado uma mensagem na JTAG UART de digitar comandos para que o usuário fique livre para escrever os comandos que deseja. As ações a serem realizadas pelo aplicativo console foram divididas em 3 partes principais: 1) Implementação com LED'S verdes, 2) Implementações com VGA pixel buffer e 3) Implementações com contador.

Essas 3 partes principais estão relacionadas com tópicos importantes que foram estudados durante o semestre no curso e entre as implementações de cada parte foram retomados vários outros aspectos estudados.

Por conta dessa visível divisão entre partes, foi mais fácil montar um cronograma para organização do tempo de realização de cada uma delas.

Para a PARTE INTRODUTÓRIA, que vem antes das implementações em si, solicitava a criação de uma comunicação entre o usuário e o processador, o que engloba o tópico de comunicação serial e conhecimentos sobre a JTAG UART , presente no simulador NIOS II, que fazia comunicação entre o Host(PC) e o processador Altera, saber que por ser comunicação serial, a UART transformava sequências de bits de serial em paralelo e vice-versa através de buffers de leitura e escrita.

Além disso, a UART contava com registradores de dados e controle, que foram usados no projeto para o desenvolvimento da comunicação com o processador.

A construção do contador englobou a parte de utilização do dispositivo I/O e utilização do temporizador, que assim como a UART, possui registradores próprios salvos em endereços de memória e possui formas de funcionamento próprias e regras para que seja utilizado.

Junto do temporizador, foi revisto as ideias de interrupção, da subrotina RTI que deveria tratar toda vez que ocorria uma interrupção.

Além das ideias de interrupção e temporizador, foi pensado em toda a lógica por trás da contagem: o que fazer quando der 60 segundos, quando der 60 minutos, quando se passavam 9 segundos e 9 minutos, e para esses 4 casos foram desenvolvidas subrotinas chamadas pela RTI.

Depois de tudo isso, se pensou em como colocar os dígitos contados no display de 7 segmentos, e para isso se criou 4 dados: V4 V3 V2 V1 salvos na memória e utilizados como base para cada dígito colocados nos displays, então os incrementos, ou decrementos e quaisquer alterações envolviam esses dados colocados nos espaços de memória.

A PRIMEIRA PARTE, que é de implementação com LED's verdes foi desenvolvida durante os dias 18/07 à 28/07.

Foi necessário um tempo maior para desenvolvê-la por conta da grande quantidade de implementações solicitadas, não pelo grau de dificuldade.

Essa parte representa o estudo dos dispositivos de entrada e saída, conhecidos como I/O, pois os LED'S verdes foram os primeiros a serem estudados, e também considerados dispositivos de comunicação paralela, ou seja, a manipulação de bits se faz de forma paralela.

No código foram desenvolvidos um total de 7 subrotinas: LED_V (que constituia de uma espécie de menu que captava o 2º dígito digitado pelo usuário para saber o caminho a seguir), cond2 e LED_V0 que juntos compuseram a parte de apagar o xx-ésimo led verde, cond1 e LED_V1 que juntos compuseram a parte de acender o xx-ésimo led verde, cond1 e cond2 foram feitos justamente para tratar o caso de acender/apagar o 8º led que na verdade constituiria um número de 9 bits(0-8 leds a serem tratados), tamanho superior ao tratado para números menores que 8 que levaram em conta apenas 8 bits.

Em seguida, foram implementadas a subrotina LED_V2 que cuidaria de deslocar todos os LED's acesos uma posição à esquerda e LED_V3 que deslocaria todos os LED's acesos uma posição à direita.

Em suma, essa parte tratou bastante de manipulação de bits, operações de deslocamento de bits (para esquerda e direita), uso de máscara para isolar bits, construção de subrotinas(formação de pilhas) e de como trabalhar com dispositivos de I/O e de forma paralela.

A SEGUNDA PARTE que é de implementação usando o VGA pixel buffer que foi desenvolvida durante os dias 29/07 à 01/08 dividindo tempo com implementações do contador.

Essa parte representa o estudo feito com gráficos e animações, porém de forma mais simples já que pedia apenas para limpar a tela do VGA pixel buffer com a cor que era representada por uma sequência de bits gerada por um dispositivo de I/O conhecido como switches.

Foi necessário o estudo de como os pixels eram salvos na memória, que eram compostos de 16 bits cada e salvos a partir de um endereço específico, havia a preocupação em saber como trabalhar com a cor e a posição de cada pixel para que em conjuntos "limpassem" a tela.

Conhecer também os endereços utilizados pelo VGA pixel buffer onde se localizavam os registradores.

Para poder compor a posição dos pixels, foram utilizados conceitos de manipulação de bits, foram necessários deslocamentos à esquerda dos dados utilizados para setar o offset de cada pixel a ser colocado na tela.

A TERCEIRA PARTE começou dentro do prazo da segunda parte e terminou no prazo de 02/08 à 06/08 e se constituía de manipulações com o contador que já tinha sido criado e funções adicionais como a criação de um alarme que deveria preencher o restante dos displays de 7 segmentos.

Para o alarme, foram adicionadas mais subrotinas chamadas pela RTI, uma delas fazia comparações a cada instante entre o número no contador e o número no alarme para ver se eram iguais.

A subrotina de comparação(contador x alarme) entrava em funcionamento apenas depois que o alarme havia sido setado, não havia sentido fazer comparações sem ter um alarme ativo.

Outras subrotinas chamadas pela RTI foram adicionadas , elas eram colocadas em funcionamento quando o número colocado no alarme fosse atingido pelo do contador e assim faziam acender os LED'S vermelhos, criavam o efeito pisca-pisca e encerravam o efeito depois de 10s.

Para que o contador ficasse ainda mais dinâmico, criou-se o efeito de pausar e despausar a contagem e para isso foi utilizado os PUSH-BUTTONS que eram dispositivos I/O de comunicação paralela mais avançada, pois contavam com mais registradores na memória que poderiam ser usados(reg de borda ,(...))

Então o RTI, além de tratar exceções causadas pelo temporizador, começou a também tratar exceções causadas pelos push-buttons.

Com as três partes em funcionamento, além da parte introdutória, começou-se a pensar em sanar bugs que poderiam ocorrer caso o usuário digitasse algum dígito inválido ou fora do contexto e assim mais código foi adicionado em cada parte na tentativa de evitar possíveis erros.

Algo que não foi citado no trabalho e que foi adicionado é o comando para terminar a execução, que “limpava” os dispositivos I/O e terminava a execução. Para isso, o usuário deve digitar 3 como primeiro dígito em alternativa ao 0,1 e 2 que já serviam para outras implementações.

Cronograma final:

Dia/Semana	Tarefa
18/07 à 28/07	Preparar ambiente para receber comandos do usuário através da UART e inicializar contador e INICIAR a implementação de funções dos leds (verdes)
29/07 à 01/08	TERMINAR implementação das funções dos leds verdes, implementar funções de vídeo e INICIAR a implementação das funções do contador
02/08 à 06/08	TERMINAR implementação das funções do contador e apresentar um relatório escrito(que deverá ser enviado no classroom)

II) Principais desafios

1- Dilema de construir o contador, pois a lógica por trás do funcionamento envolvia vários casos , o que trouxe como consequência a criação de 4 subrotinas extras que teriam que atuar em conjunto: ZERAR_S, ZERAR_M, MIN_1,MIN_60, além disso, outro desafio foi pensar em como colocar os dígitos no display de 7 segmentos, tendo que criar dados na memória e trabalhar com eles, ora usando load, ora store.

2- Dilema da organização das subrotinas.

Teve muito a se pensar em como o programa iria do main para cada subrotina dependendo de cada dígito digitado pelo usuário e posteriormente teriam que voltar na parte em que o programa libera ao usuário para digitar novos comandos.

Para isso, foi criada uma seção de LABEL's no programa principal a fim de gerenciar a chamada das subrotinas e o retorno para a parte correta que o código deveria continuar.

3-Dilema do funcionamento do alarme

O problema em si não foi setar o alarme no display de 7seg, mas sim o momento em que o contador atingia o valor do alarme, pois precisava acender os LED's vermelhos, fazer efeito pisca-pisca, fazer esse efeito durar 10s e ainda por cima isso deveria ocorrer a cada 0,5s.

Porém, até essa parte do programa o temporizador fazia interrupções a cada 1s.

Então, teve que mudar interrupções de 1s para 0,5s e criar variáveis para analisar se era a 1ª interrupção ou a 2ª, para direcionar corretamente o momento em que deveria tratar o contador.

Além disso, muitas subrotinas chamadas pela RTI tiveram que ser adicionadas para cumprir com todas essas solicitações.

III) Conclusão

Muito já se foi falado na introdução, mas observando o trabalho como um todo, ele ajudou a relembrar e tirar possíveis dúvidas a respeito de vários tópicos que foram tratados no decorrer do semestre: desde manipulação e operação de bits, trabalhar com diversos dispositivos que funcionam seja de forma paralela, seja de forma serial, em criar subrotinas e estruturá-las junto das pilhas, em trabalhar com a tela VGA com gráficos, entre outros.

Permitiu ir mais profundo em entender a linguagem Nios II, que tem arquitetura RISC e como o processador Altera funciona.

Todo o decorrer do curso no semestre, desde os laboratórios até a realização desse trabalho foi possível graças ao simulador online do Nios II, que contém os mais variados dispositivos vistos e mais funções que nem foram abordadas na disciplina.

De forma geral, existiram conteúdos mais fáceis de se aprender e outros mais difíceis, longos, cansativos.

O docente responsável pela disciplina organizou em um cronograma o conteúdo de cada semana e isso foi bem útil para ter uma base e não se perder com tanto conteúdo.

Semanalmente, existiam monitorias para que os alunos pudessem tirar dúvidas a respeito dos conteúdos de cada semana e isso é um ponto positivo, pois conteúdos novos sempre geram receio e questionamentos.

Além disso, nesses encontros o docente ia além de tirar dúvidas, mas complementando o conteúdo de cada semana com novos exercícios que estimulavam a programação e o pensar.

Para cumprir com o rendimento do que era visto a cada semana, deveria se fazer um laboratório, todos eles foram úteis para concretizar o entendimento e auxiliaram bastante a entender o projeto final.

Um ponto a se pensar e que poderia ser adotado seria organizar a agenda de maneira mais livre, com assuntos que levassem 2 ou 3 semanas dependendo da dificuldade dos alunos e da quantidade de conteúdo e não “empacotar” os assuntos colocando obrigatoriamente 1 a cada semana, sem um possível tempo para organizar os pensamentos e absorver cada tópico da matéria.

Dessa forma, os alunos teriam mais tempo para resolver os laboratórios e pensar nos tópicos estudados.