

UNIVERSIDADE FEDERAL DE MINAS GERAIS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Organização de Computadores I
Trabalho Prático II – Processador Nibble (Verilog)

Gabriel Lima Canguçu
João Marcos Couto
Pedro Rios Bustamante
Thiago Martin Poppe

Professor: Daniel Macedo

Belo Horizonte
1 de Dezembro de 2018

Sumário

1	Introdução	2
2	Desenvolvimento da Proposta e Solução dos Problemas	2
3	Módulos	2
3.1	Memória de programa (ROM)	2
3.2	Decodificador de instrução	2
3.3	Memória de dados (RAM)	2
3.4	Multiplexadores	2
3.5	Módulo complemento a 2	3
3.6	Somador completo (8 bits)	3
3.7	Registradores de Carga Paralela	3
3.8	Módulo de Controle Central	3
4	Diagrama de Símbolos	3
5	Testes	4
5.1	Testes	4
5.2	Testes para o Processador	7
6	Conclusão	7

1 Introdução

A finalidade do trabalho é a implementação do Niblle, um processador de 8 bits, cujo conjunto de instruções é reduzido. Esse processador possui apenas 6 instruções: ADD, ADDP, SUB, SUBP, END e MDUMP. Para implementá-las usamos o iverilog. Para criar o diagrama de símbolos e unir os módulos utilizamos o Quartus. Sendo usados para isso os seguintes módulos:

1. Memória de programa (ROM)*;
2. Decodificador de instrução;
3. Memória de dados (RAM)*;
4. Multiplexadores;
5. Módulo complemento a 2;
6. Somador completo (8 bits);
7. Registradores de carga paralela (PC, ADDR, X, OP, Y e R);
8. Módulo de controle central.

Sendo que os módulos 1 e 3 foram fornecidos, portanto não precisaram ser desenvolvidos.

2 Desenvolvimento da Proposta e Solução dos Problemas

A implementação de cada módulo envolvia uma dificuldade diferente e habilidades diferentes referentes à programação em verilog. A tarefa foi dividida em partes criando os módulos separadamente e finalmente conseguindo a solução para cada um deles separadamente. Só depois de resolvidas as respectivas implementações o processo passou a ser criar o projeto no Quartus e identificar como integrar cada uma das partes já desenvolvidas no projeto. Pudemos observar que a implementação mais simples e que evitaria uma maior quantidade de erros seria a sem pipeline e de implementação uniciclo.

Para os testes na implementação dos muxs, simplesmente a compilação não era o suficiente, foi necessário printar diferentes resultados para testes que também tivemos que desenvolver, utilizando do comando \$monitor e do comando para descobrir as saídas que cada mux obtinha, variando as entradas, dessa forma era possível identificar o erro e fazer as alterações necessárias.

3 Módulos

A seguir temos uma breve descrição módulo por módulo de como esses foram implementados:

Nota: os módulos 1 e 3 não serão descritos pois foram dados.

3.1 Memória de programa (ROM)

Módulo não implementado.

3.2 Decodificador de instrução

O módulo decodificador é responsável por separar o conteúdo da entrada em 4 partes, um código de endereço de operação, um endereço de memória e dois operandos de 8 bits (X e Y).

3.3 Memória de dados (RAM)

Módulo não implementado.

3.4 Multiplexadores

Os multiplexadores são dispositivos que selecionam as informações de dados de duas ou mais fontes de dados em um único canal, ou seja, realiza uma convergência das informações dadas em

uma saída. foram implementados multiplexadores 3:1 e 2:1, devido à necessidade dos dois pois o primeiro precisa de um bit de seleção a mais que o segundo.

3.5 Módulo complemento a 2

O complemento de dois é um cálculo simples, quando se usa papel e lápis, e mais simples ainda em questão de implementação. O complemento de dois consiste somente na inversão de bits do número binário de entrada e somando um tem-se a saída desejada. O módulo segue essa ideia.

3.6 Somador completo (8 bits)

O módulo da implementação de um full adder de 8 bits poderia ter sido feito de duas formas. Poderíamos ter usado adders de dois bits como intermedio entre as entradas e saídas, mas optamos por utilizar uma implementação usando somente ands, dispensando dessa forma os adders de dois bits, devido a uma maior facilidade de implementação nessa forma. O somador completo também cuida dos casos de overflow, setando o overflow para 1, indicando o erro para o controlador central.

3.7 Registradores de Carga Paralela

Os registradores de carga paralela são: PC, INST, ADDR, Y, X, OP, X e R. Não precisamos de um enable por o processador usado é unicycle. A mudança do dado de saída é engatilhada na subida do clock.

3.8 Módulo de Controle Central

Devido a não utilização do pipeline o controle central não trabalha exatamente com as instruções FETCH, DECODE, EXECUTE/STORE. Apesar disso ele consegue executar as funções referentes às instruções de outras formas. Esse é o módulo mais importante pois ele controla todos os outros módulos, ou seja, a sua implementação interfere em todo o sistema.

4 Diagrama de Símbolos

Foi necessária a criação de um diagrama de símbolos para o processador, segue a imagem do diagrama criado:

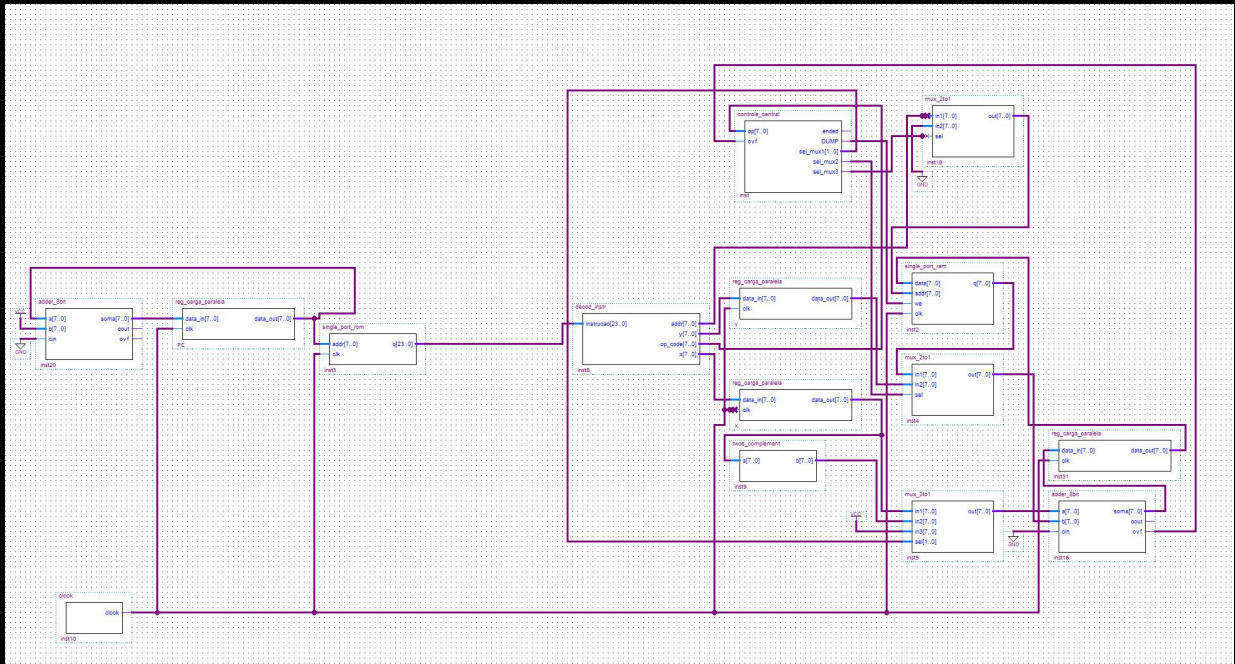


Figura 1 Diagrama do Processador

5 Testes

5.1 Testes

Alguns dos casos de testes utilizados para os módulos seguem abaixo:

1. Decodificador de instrução

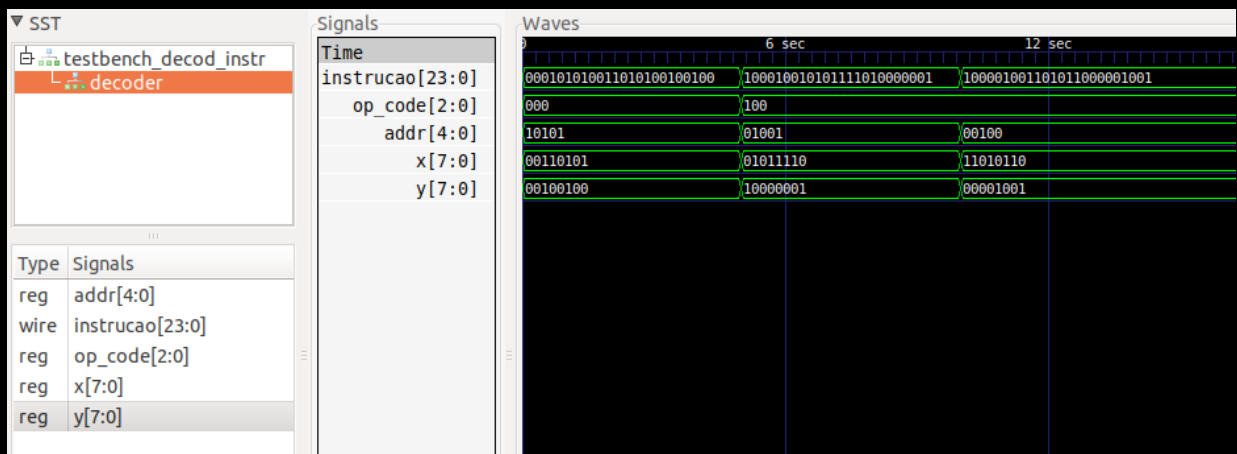


Figura 2 Testes para o Decodificador

2. Multiplexadores



Figura 3 Testes para MUX 2:1

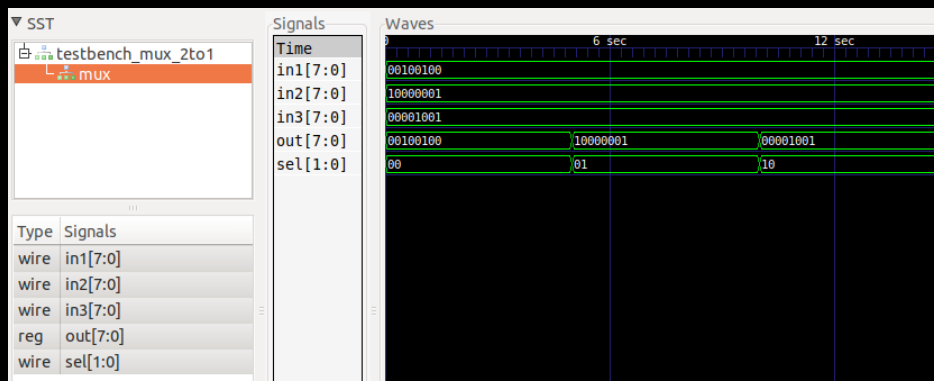


Figura 4 Testes para MUX 3:1

3. Módulo complemento a 2

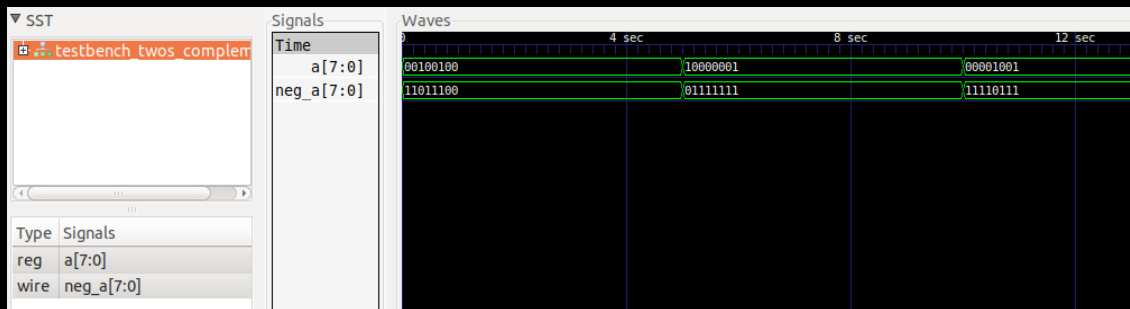


Figura 5 Testes para o módulo de complemento a 2

4. Somador completo

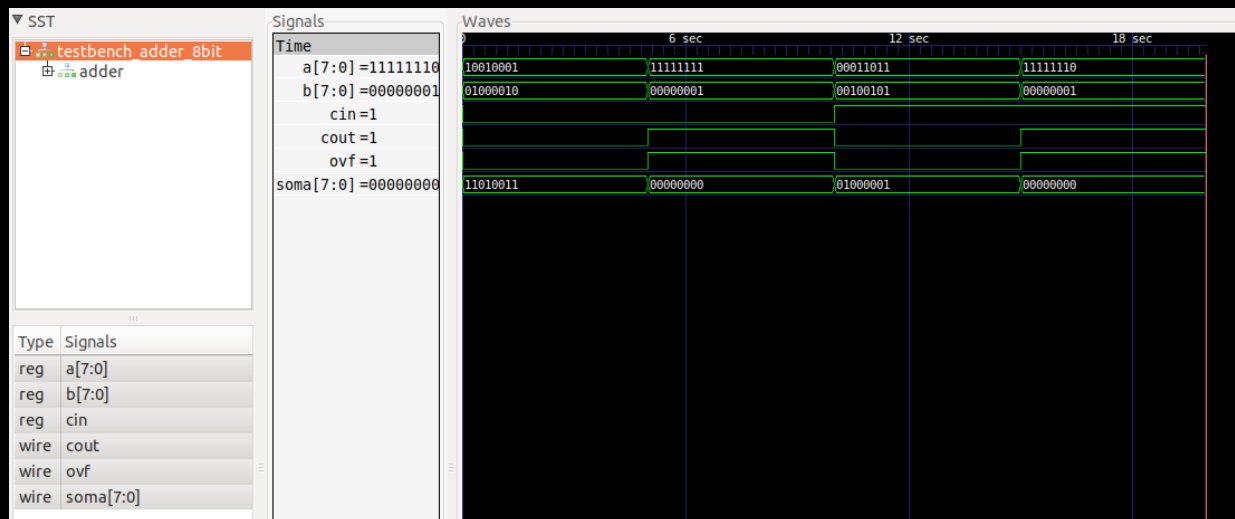


Figura 6 Testes para o Somador

5. Registradores de carga paralela

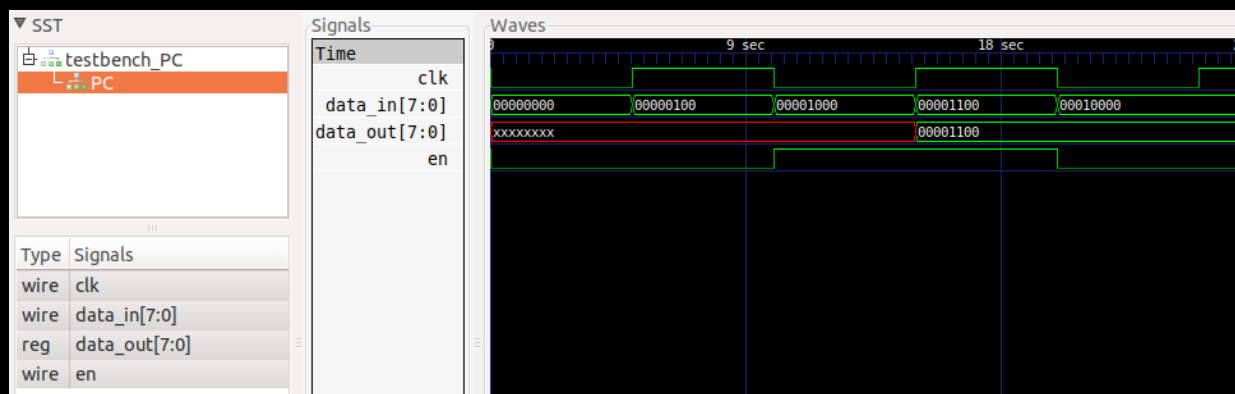


Figura 7 Testes para o PC

6. Módulo de controle central

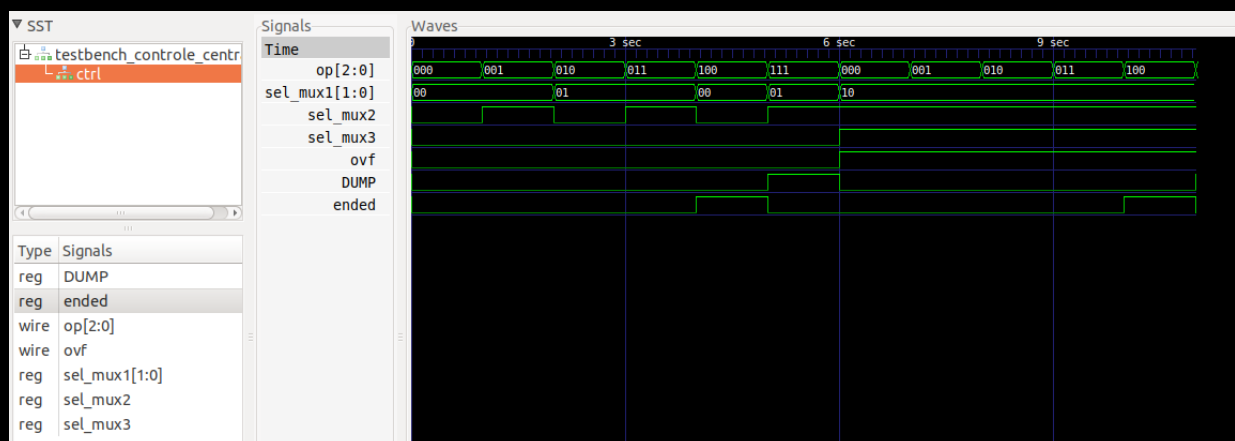


Figura 8 Testes para o Controle Central

5.2 Testes para o Processador

Não conseguimos elaborar testes a tempo da entrega. Porém todos os módulos estão plenamente funcionais e, se compilados em conjunto num mesmo projeto, não ocorrem quaisquer erros.

6 Conclusão

O processador completamente integrado foi implementado no Quartus Prime II e conseguiu-se compilar o código integralmente e sem erros, no entanto não conseguimos elaborar testes para atestar a funcionalidade plena do processador. Todavia, nesta documentação demonstramos que individualmente todos os módulos estão plenamente funcionais.