

Decisões do Projeto

Para a implementação dessa ISA busquei fazer um projeto simples, mirando em deixá-lo fácil de ser compreendido, limpo e conciso. Para isso optei por utilizar apenas um formato de instrução, dessa maneira, todos os bits sempre estarão alinhados e nenhuma lógica adicional é necessária para selecionar os bits utilizados.

Dos 24 bits para a instrução, gostaria de manter uma quantia balanceada de tudo. Sendo assim, optei por deixar 4 bits para cada registrador, dessa forma posso selecionar 16 registradores diferentes no meu processador e gastar apenas 12 bits para isso. A respeito do OPCODE, utilizei 4 bits também, usar mais que isso deixaria o “Imediato” muito pequeno, tornando difícil de programar saltos, fora isso, para o propósito de uma ISA simples, 16 comandos já são mais do que o suficiente, permitindo incluir todas as instruções lógicas, aritméticas, de desvio, saltos e memória necessárias.

Foi decidido manter o primeiro registrador como o número 0 fixo, pois isso facilita muito a programação. O segundo registrador é um ponteiro para a memória, para que seja possível trabalhar com funções, o último é o registrador que deve guardar o endereço em que uma função chamada deve retornar.

Presumi que 3 registradores para a passada de parâmetros seja um número decente em relação a quantidade total, na mesma linha 4 registradores temporários foram definidos e 6 registradores salvos. Assim, uma quantia balanceada de cada um deles está disponível para o usuário.

As instruções escolhidas foram as seguintes:

- Aritmética e lógicas:
 - AND
 - OR
 - ADD
 - ADDI
 - SUB
 - SSLI
- De salto e decisão:
 - JAL
 - JALR
 - BEQ
 - BNE
 - BLT
 - BGE
- De memória:
 - LW
 - SW
- De parada:
 - HALT
 - RESET

As funções escolhidas permitem que sejam realizados loops, chamadas de função, desvios condicionais e etc...

Funcionamento das Unidades

1. PROGRAM COUNTER

Essa é a unidade responsável por cuidar do PC, consiste em um registrador que armazena o local atual em que o programa está. No caso padrão, a cada ciclo de clock este registrador é atualizado para $PC = PC + 1$, porém existe um MUX controlando quanto será adicionado, as opções são a padrão, receber $PC + \text{Imediato}$ ou $Rs1 + \text{Imediato}$.

Uma variável controla se o PC poderá ser atualizado ou não, ela é alterada somente pelo comando HALT, que finaliza a execução de um programa. Essa unidade também possui um RESET, que poderá ser ativada manualmente pelo simulador ou pelo comando RESET.

2. MEMÓRIAS

A memória ROM ficou responsável por guardar as instruções que executarão no processador, sendo assim, ela possui 24 bits de endereço e de dados. Já a memória RAM guarda os dados e permite o usuário realizar a escrita e leitura, está possui 24 bits de endereço, porém, 32 bits de dados. Ambas as memórias são componentes padrão do logisim e não são endereçadas byte a byte.

3. BANCO DE REGISTRADORES

O banco de registradores possui 16 registradores que são possíveis de acessar, sendo o primeiro deles apenas uma constante "0" de 32 bits. Dentro do banco existem 2 MUX, que controlam quais registradores estão sendo lidos e 1 DEMUX responsável por escolher qual será escrito. Uma variável de controle toma conta da escrita em RD e esse dado a ser escrito dependerá do comando, por isso é controlado por um MUX. Analogamente ao PC, possui um RESET.

4. ULA

A ULA possui um código que decide qual operação será realizada e um código que diz se ela deverá realizar a operação com a informação vinda do banco de registradores ou com o imediato. Dentro dela existem 5 operações que suprem as necessidades do meu processador. A ULA realiza soma, subtração, shift lógico e faz ANDs e ORs.

5. UNIDADE DE CONTROLE E BRANCHES

Por último, a unidade de controle é responsável por administrar quais pinos serão ligados e desligados, ela faz isso através dos 4 bits vindos do OPCODE, calculando então quais variáveis deveram estar ativas. O cálculo foi feito através da elaboração de uma tabela que associava os comandos com os pinos, após isso eu retirei os Maxitermos e Mintermos mais apropriados para cada situação, simplifiquei-os e montei a Unidade de Controle.

Branches foi o nome que eu escolhi para o componente que faz as comparações lógicas maior, menor, igual etc... Ela recebe qual branch está sendo realizada e devolve 1 ou 0, indicando se o pulo deve ou não ser tomado

Listagem dos Componentes

- PC: Um registrador de 24 bits
- ROM: Memória de instruções com 24 bits de endereço e 24 bits de dados
- RAM: Memória de dados com 24 bits de endereço e 32 bits de dados
- Banco de Registradores: 16 registradores com 32 bits cada um
- ULA: Unidade de lógica e aritmética que trabalha com 32 bits
 - Soma
 - Subtração
 - ShiftLeft
 - AND
 - OR
- UNIT CONTROL: Unidade de controle responsável por controlar as seguintes variáveis do processador
 - W_REG: Decide se algo será escrito no RD ou não
 - W_MEM: Decide se algo será escrito na memória de dados
 - W_PC: Decide se algo será escrito no PC
 - IMM_ULA: Decide se a ula irá usar o imediato ou Rs2 para a operação
 - OP_ULA: Decide qual tipo de operação a ULA deverá fazer
 - B: Diz se o comando executado é do tipo Branch (Ativa o componente “Branches”)
 - B_IN: Diz qual comando do tipo Branch está sendo executado
 - D_RG: Decide qual informação será escrita no banco de registradores
 - PC_M: Decide qual tipo de acréscimo o PC receberá (exemplo: $Pc + 1$, $Pc + \text{Imediato}$, $Rs1 + \text{Imediato}$...)
 - RESET: Reseta todos os registradores e o PC para zero
- BRANCHES: Lógicas que operam sob 32 bits para verificar se o pulo do comando Branch deverá ser tomado ou não