

Processador Calculadora

Design de Computadores

João V. Rodrigues, Jose F. Cruz

Arquitetura

A arquitetura escolhida para esse projeto foi a arquitetura Registrador-Memória, onde as operações acontecem entre registradores e valores salvos na memória. É uma arquitetura intermediária, que não apresenta alta complexidade de código VHDL, assim como uma arquitetura que é suficiente para o projeto da calculadora

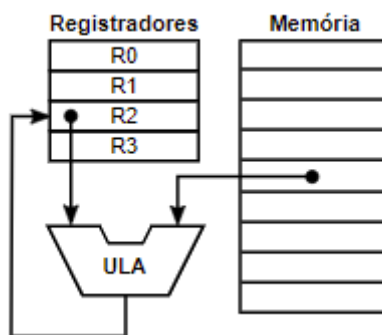


Imagem 1: Esquema simplificado da arquitetura Registrador-Memória

Instruções

Nesse projeto o endereçamento é feito de forma direta, ou seja, o endereço é passado diretamente na instrução, uma das formas mais simples de endereçamento.

| Instrução | OpCode | Descrição |
|-----------|--------|---|
| LOAD | 00000 | Carrega a posição da memória no registrador |
| STORE | 00001 | Salva o conteúdo do registrador na memória |
| JMP | 00010 | Jump incondicional |
| JE | 00011 | Jump se igual |
| JLT | 00100 | Jump se menor que |
| JSR | 00101 | Jump para um endereço |
| RET | 00110 | Jump devolta |
| Soma | 00111 | Soma simples reg/mem |
| Sub | 01000 | Subtração simples reg/mem |
| Soma im | 01001 | Soma com imediato |
| Sub im | 01010 | Subtração com imediato |
| Soma car | 01011 | Soma com carry |
| Sub bor | 01100 | Subtração com borrow |
| Mul | 01101 | Multiplicação |
| Div | 01110 | Divisão |
| AND | 01111 | AND lógico |
| OR | 10000 | OR lógico |
| NOT | 10001 | NOT lógico |
| XOR | 10110 | XOR lógico |
| MOV | 10011 | Move valor imediato para <u>registrador</u> . |

Nesse projeto o endereçamento é feito de forma direta, ou seja, o endereço é passado diretamente na instrução, uma das formas mais simples de endereçamento.

A palavra de instrução para esse projeto contém 16 bits, descrito na tabela 2:

| OpCode | Registrador | Endereço |
|-----------|-------------|----------|
| 5 bits | 3 bits | 8 bits |
| [23 – 19] | [18 – 16] | [8 – 0] |
| MSB(b23) | | LSB(b0) |

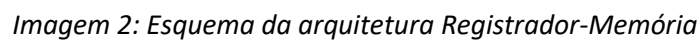
Tabela 2: Formato da instrução

Onde:

- 5 bits são referentes ao OpCode, possibilitando até 32 instruções;
- 3 bits para os descrever 8 registradores;
- 8 bits que descrevem 265 posições na memória

| Ponto de Controle | | Descrição |
|-------------------|--------|--|
| JumpMux | 1 bit | Ativado quando uma instrução vinda da ROM chega solicitando um salto na memória. |
| Mux | 1 bit | Controla o MUX que recebe os dados da memória de instrução; |
| Hab. Escrita Reg | 1 bit | Ativa a escrita nos registradores no banco de registradores; |
| Write | 1 bit | Ativa a leitura na memória RAM; |
| Read | 1 bit | Ativa a escrita na memória RAM; |
| Operação ULA↓ | 3 bits | |
| Soma | 000 | Soma os dados da entrada A com a entrada B |
| Subtração | 001 | Subtrai os dados da entrada A com a entrada B |
| AND | 010 | |
| OR | 011 | |
| XOR | 100 | |
| NOT | 101 | |

Tabela 3: Formato dos pontos de controle



- Com a arquitetura Harvard, não precisaremos alocar nada para a memória ROM
- A RAM ficará entre os endereços 0 a 63;
- As chaves de 0 a 7 estão na posição 64;
- As chaves 8 e 9 estão na posição 65;
- O botão 0 ocupa a posição 66;
- O botão 1 ocupa a posição 67;
- Os displays ocupam posições agrupadas, onde:
 - Os displays 0 e 1 estão na posição 128;
 - Os displays 2 e 3 estão na posição 129;
 - O display 4 está na posição 130;
 - O display 5 está na posição 131;

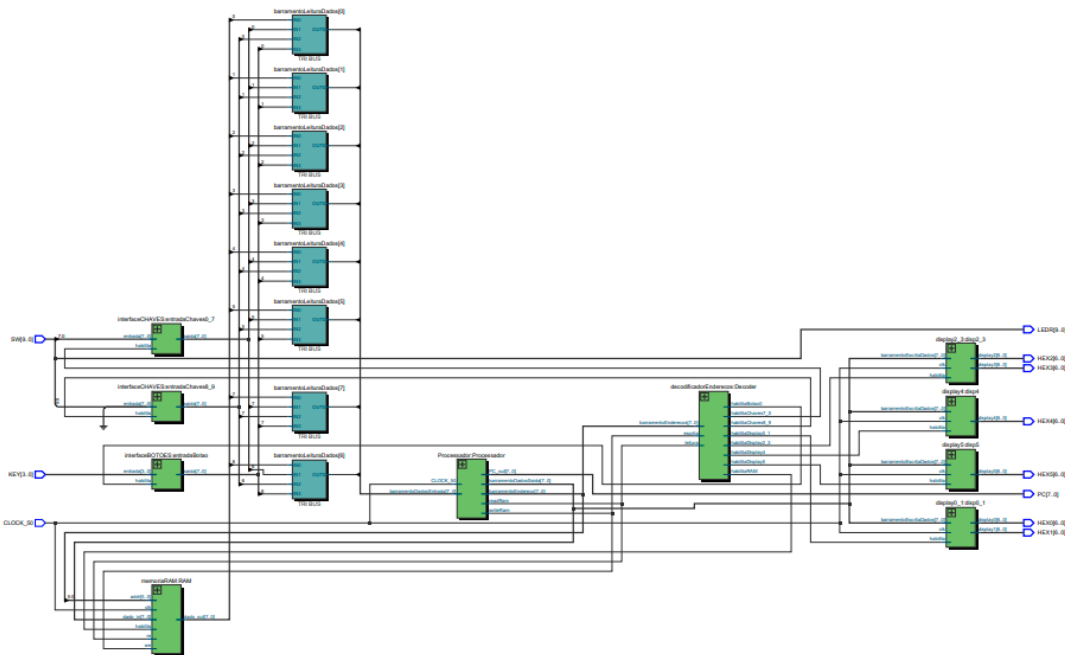


Imagem 3: Diagrama de blocos da calculadora desenvolvida.

Manual

A calculadora funciona para números de até 16 bits, da seguinte forma:

Entrada de valores:

A chave 9 indica se a variável de entrada é A ou a B, sendo A e B as variáveis a serem calculadas. Chave 9 igual a zero indica variável A, e chave 9 igual a um indica variável B;

A chave 8 indica se os números de entrada são referentes a parte mais significativa ou menos significativa do número de entrada. Chave 8 igual a zero indica a parte menos significativa e chave 8 igual a um indica a parte mais significativa o número.

As chaves de 7 a 0 são a entrada de número de 8 bits, podendo ser de até 16, baseada na instrução da chave 8.

Operações

As operações ocorrem através dos botões:

O botão 0 armazena os números de entrada;

O botão 1 realiza a operação com os números inseridos;

Modo de usar

Insira os 8 bits menos significativos da variável A e aperte o botão 0 para armazená-los;

Insira os 8 bits mais significativos da variável A e aperte o botão 0 para armazená-los;

Repita o processo para a variável B, mudando a chave 9 de zero para 1;

Ao fim do armazenamento das duas variáveis, insira o opCode da operação de deseja realizar. Os opCodes de operação podem ser encontrados na *tabela 4*.

Após inserir o opCode da operação, aperte o botão 1 para executar a operação;

O resultado será exibido nos displays de sete segmentos.

| Operação | OpCode | Descrição |
|---------------|--------|---------------------------------------|
| Soma | 00111 | Soma simples reg/mem |
| Subtração | 01000 | Subtração simples reg/mem |
| Multiplicação | 01101 | Multiplicação |
| Divisão | 01110 | Divisão |
| AND | 01111 | AND lógico |
| OR | 10000 | OR lógico |
| NOT | 10001 | NOT lógico |
| XOR | 10110 | XOR lógico |
| MOV | 10011 | Move valor imediato para registrador. |

Tabela 4: OpCode das operações.