

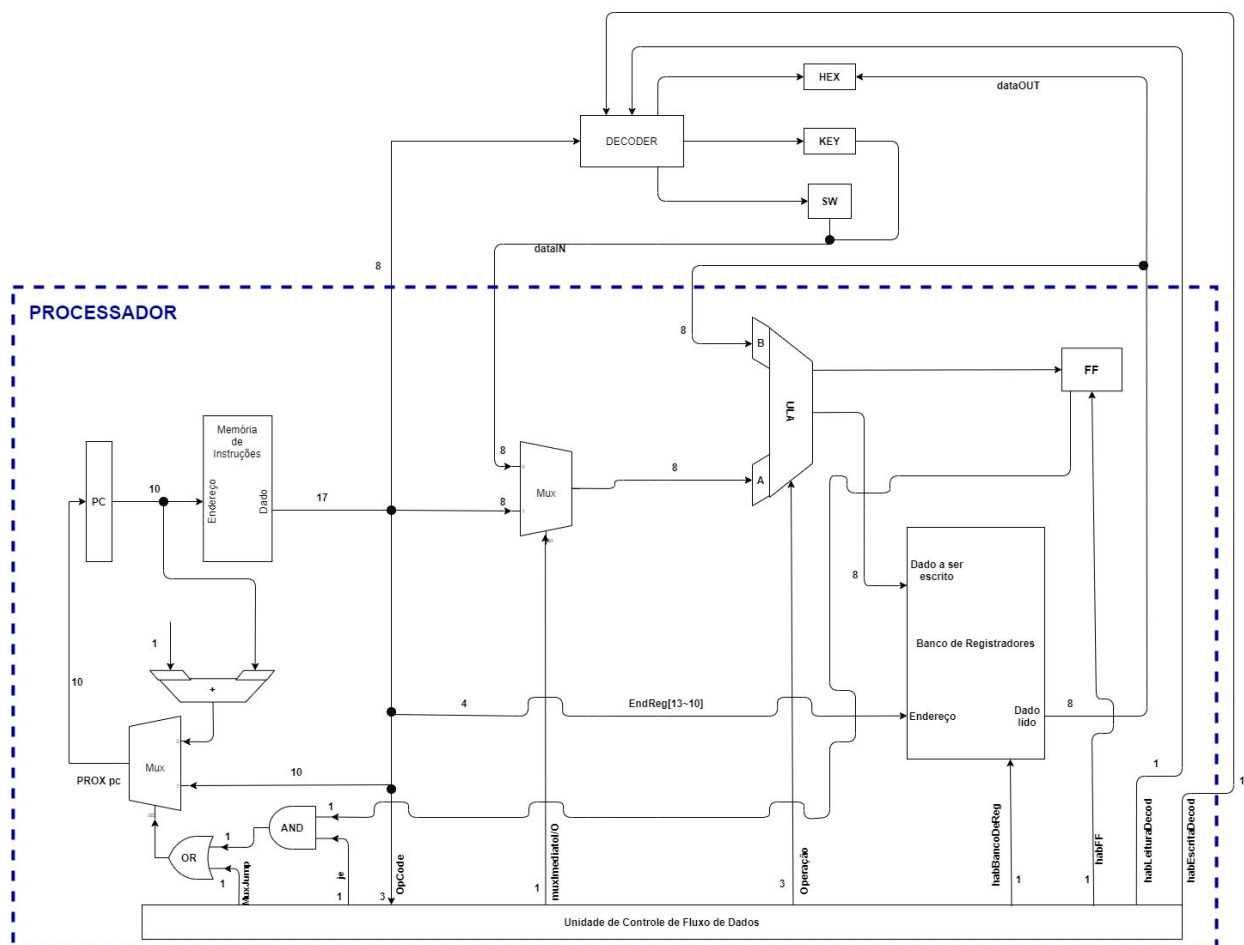
Relógio Utilizando um Processador Personalizado

Entrega Intermediária

Matheus Pellizzon, Pedro Paulo Telho, Pedro Ramos

Arquitetura do Processador e Fluxo de Dados:

A arquitetura escolhida para o processador foi a de registrador-memória com algumas modificações para que possa ser possível escrever e ler nos próprios registradores, conforme o diagrama abaixo:



Mesmo que a arquitetura de acumulador seja mais simples de implementar em relação ao hardware, ela exige mais atenção na implementação do software do relógio.

Dessa forma a arquitetura registrador-memória apresenta uma facilidade maior nesse aspecto e por isso optamos por ela.

Formato das Instruções:

As instruções seguem o seguinte formato:

OpCode	Endereço do Registrador	Reservado
3 bits [16~14]	4 bits [13~10]	10 bits [9~0] ROM [7~0] Imediato

Exemplos:

add: 01001110000000001

jmp: 001XXXX1000000000

Vetor de bits do ponto de controle:

Vetor de Controle							
9	8	7	6	5 - 3	2	1	0
muxJump	je	muxImediato/O	habEscritaDeReg	Operação	habFF	load	store

Total de Instruções e sua Sintaxe:

Instrução	Descrição	OpCode
cmp	Habilita Eq se A==B na ULA	000
jmp	Desvia o PC	001
add	Ry <= Rx + Imediato	010
sub	Ry <= Rx - Imediato	011
getio	Rn <= dataIN	100
display	dataOUT <= Rn	101

je	Desvia o PC se Eq == 1	110
mov	Rx <= Imediato	111

Ou seja, no total existem 8 instruções necessárias para a implementação do relógio.

Listagem dos Pontos de Controle e sua Utilização:

- **muxJump:** define se o Program Counter deve ser desviado ou não. Quando o sinal desse ponto de controle é 0 o próximo valor do PC é o valor anterior mais 1. Já quando o MuxJump assume valor 1 o PC é desviado para o endereço da ROM que chega pelo barramento vermelho no esquema do processador;
- **je:** define se o PC deve ser desviado ou não. Quando existe uma comparação o sinal je é acionado e se o resultado da comparação é verdadeiro ocorre o desvio.
- **muxImediato/O:** esse ponto controla se uma operação será feita usando um dado que vem da RAM ou se utilizará o imediato;
- **habEscritaDeReg:** define se o dado de saída da ULA será escrito em algum dos registradores;
- **Operação:** define qual operação a ULA realizará com os dados recebidos;
- **habFF:** ponto de controle utilizado para habilitar o flip flop que guarda a saída da comparação realizada na ULA;
- **load:** ponto de controle usado para habilitar a leitura no decoder;
- **store:** ponto de controle responsável por habilitar a escrita no decoder.

Rascunho do Mapa de Memória:

- Total de posições de memória que o processador é capaz de endereçar: 128;

- Quantidade de posições de memória que os switches ocupam: 10 [0x0~0x9];
- Quantidade de posições de memória que os botões ocupam: 4 [0xA~0xD];
- Quantidade de posições de memória que displays hexadecimais ocupam: 6 [0xE~0x13].

Endereços: [Início~Fim]