

# Uma CPU simples para fins didáticos

WILIAN SOARES LACERDA

UFLA – Universidade Federal de Lavras  
DCC – Departamento de Ciência da Computação  
Cx. Postal 37 – CEP 37.200-000 Lavras (MG)  
lacerda@ufla.br

**Resumo:** Este artigo apresenta o desenvolvimento de uma CPU (Unidade Central de Processamento) com uma arquitetura simples para utilização em ensino na disciplina de Arquitetura de Computadores. A CPU proposta contém todos os elementos básicos de um processador digital, permitindo o estudo do seu funcionamento completo.

**Palavra Chave:** CPU, arquitetura de processador

## 1 Introdução

Uma unidade central de processamento, ou CPU, é o principal componente dos computadores digitais. É a parte do computador responsável pela manipulação e transformação dos dados ou informações [4].

Desde o primeiro computador digital, até hoje, é utilizado no projeto do processador a topologia denominada arquitetura de von Newmann. O que diferencia a arquitetura dos atuais processadores são a tecnologia de fabricação, velocidade de processamento, complexidade do projeto, dentre outras [3].

Este artigo propõe o projeto de uma CPU simples, que contém as características básicas de um processador digital [2]. Tem como objetivo servir de exemplo para ensino em curso de graduação em Ciência da Computação na Universidade Federal de Lavras, em especial para a disciplina de Arquitetura de Computadores.

## 2 A unidade central de processamento

A CPU proposta neste artigo possui a seguinte estrutura externa básica, mostrada na Figura 1.

Os sinais de entrada e saída da CPU proposta possuem as seguintes funções e características:

- D7..D0 – Barramento de dados externo (8 bits)
- A7..A0 – Barramento de endereço (8 bits)
- /MEM – sinal de habilitação da memória, ativa em nível zero
- /IO – sinal de habilitação da porta de entrada/saída, ativa em nível zero

- /WR – indica ciclo de escrita em porta ou memória, ativa em nível zero
- /RD – indica ciclo de leitura em porta ou memória, ativa em nível zero
- CLOCK – sinal de sincronismo
- /RESET – sinal de inicialização, ativo em nível zero

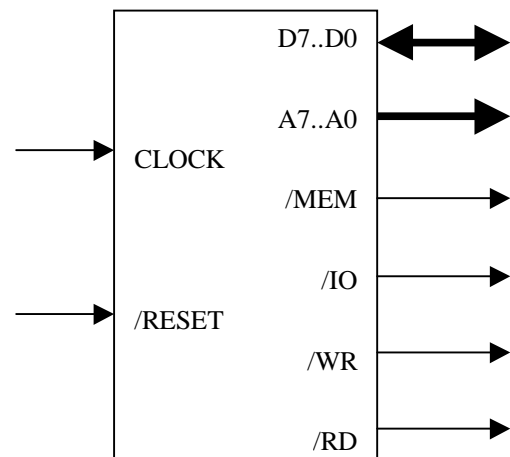
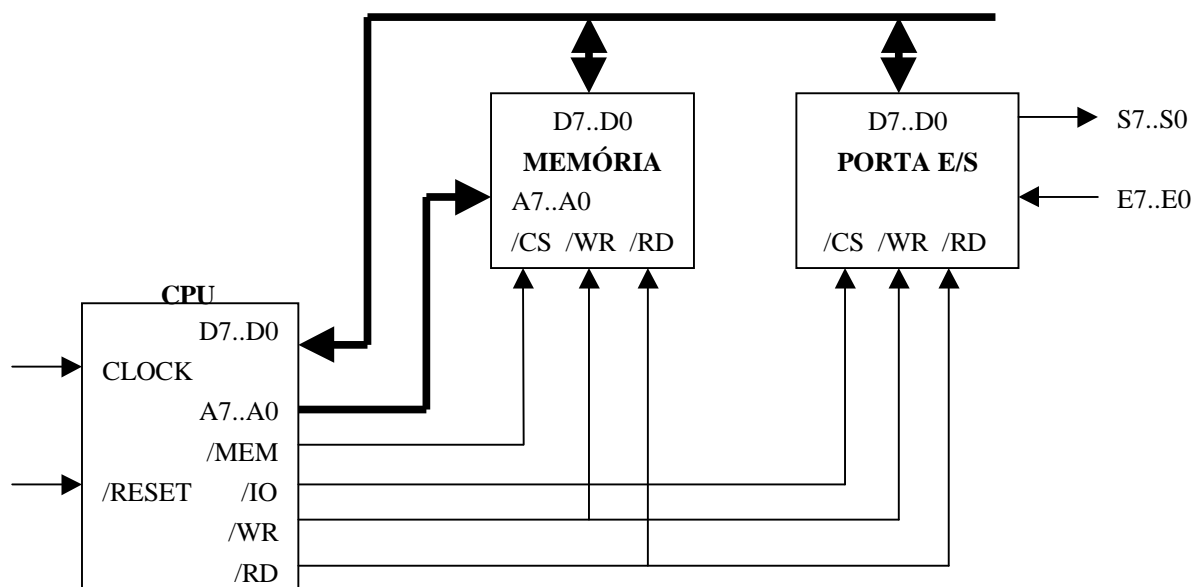


Figura 1: CPU

A CPU não é o único componente de um computador. Existem circuitos auxiliares que em conjunto com a CPU formam um sistema de computador. Dentre estes circuitos, existe a memória e as portas de entrada/saída que são conectadas à CPU como é mostrado na Figura 2.





**Figura 2:** Um sistema de computador com memória e porta entrada/saída

A memória é responsável pelo armazenamento das instruções (ou comandos do processador) e dos dados. Uma memória, em geral, tem uma grande capacidade de armazenamento. Cada informação é armazenada em uma posição da memória que é identificada por um endereço (conjunto de bits). Através deste endereço, é possível recuperar a informação armazenada na memória.

As portas de entrada/saída são circuitos que permitem a troca de informações entre o computador e o ambiente externo. As portas de entrada enviam dados gerados por circuitos externos para a CPU, que por sua vez pode armazená-los na memória. As portas de saída permitem que a CPU transmita dados para circuitos externos.

Para permitir a comunicação da CPU com os circuitos auxiliares, a própria CPU gera os sinais responsáveis pelo controle da memória e das portas de E/S. Os sinais de controle da memória são:

- /MEM – habilita a memória para leitura ou escrita, ativo em zero.
- /RD – habilita leitura, ativo em zero.
- /WR – habilita escrita, ativo em zero.

Os sinais de controle da porta de entrada/saída são:

- /IO – habilita leitura ou escrita da porta, ativo em zero.
- /RD – habilita leitura, ativo em zero.
- /WR – habilita escrita, ativo em zero.

### 3 Componentes da CPU

Para facilitar o projeto e estudo da CPU, ela é dividida internamente em duas unidades fundamentais:

- Unidade de dados
- Unidade de controle

Estas unidades componentes funcionam em conjunto e conectadas entre si. Elas podem ser visualizadas na Figura 3, onde se observa a ligação entre elas.

A unidade de dados possui as seguintes funções:

- Cálculos aritméticos
- Funções lógicas
- Manipulação de dados
- Armazenamento temporário de dados
- Recebimento de dados
- Envio de dados

A unidade de controle possui as seguintes funções:

- Geração dos sinais de controle da unidade de dados
- Geração dos sinais de controle externos
- Sincronização dos sinais de controle
- Inicialização do sistema
- Geração do endereço de memória
- Busca e armazenamento das instruções e dados

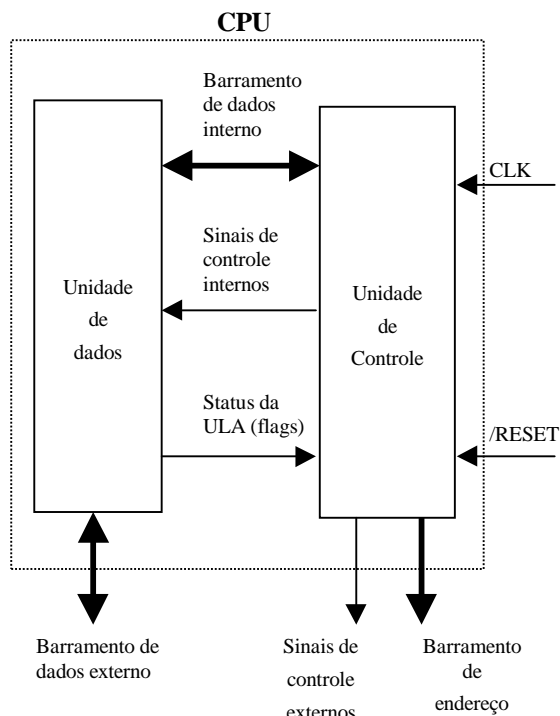


Figura 3: Unidades da CPU

### 3.1 Unidade de dados

A unidade de dados é composta de ULA (unidade lógica aritmética), registradores, multiplexadores, buffers tri-state e barramentos, conforme mostrado na Figura 5.

A ULA é a responsável pelas operações lógicas e aritméticas da CPU. A ULA proposta é mostrada na Figura 4 com os sinais de entrada e saída. Ela é capaz de realizar as operações descritas na Tabela 1, conforme o código da operação gerado pela unidade de controle.

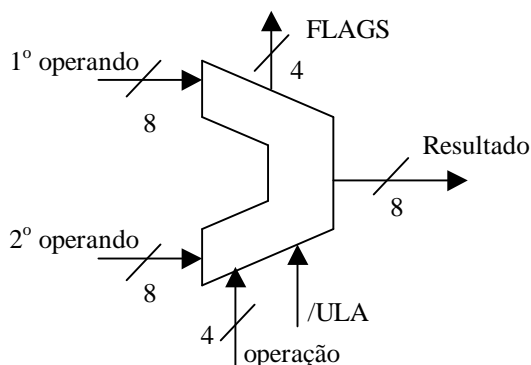


Figura 4: ULA

Tabela 1: Código de operação da ULA

Código Op. (binário)	Resultado	Símbolo
0000	Não faz nada	NOP
0001	Soma (1º op. + 2º op.)	ADD
0010	Subtrai (1º op. - 2º op.)	SUB
0011	1º op. AND 2º op.	AND
0100	1º op. OR 2º op.	OR
0101	1º op. XOR 2º op.	XOR
0110	Complementa 1º op.	NOT
0111	1º op. + 1	INC
1000	1º op. - 1	DEC
1001	Deslocamento 1º op. para direita de 1 bit	SHR
1010	Deslocamento 1º op. para esquerda de 1 bit	SHL
1011	Nega o 1º op.	NEG

A ULA é composta internamente por circuitos lógicos somadores, subtratores, deslocadores, e inversores para a realização da operação selecionada. Ela possui os seguintes sinais de entrada e saída:

a) Entradas:

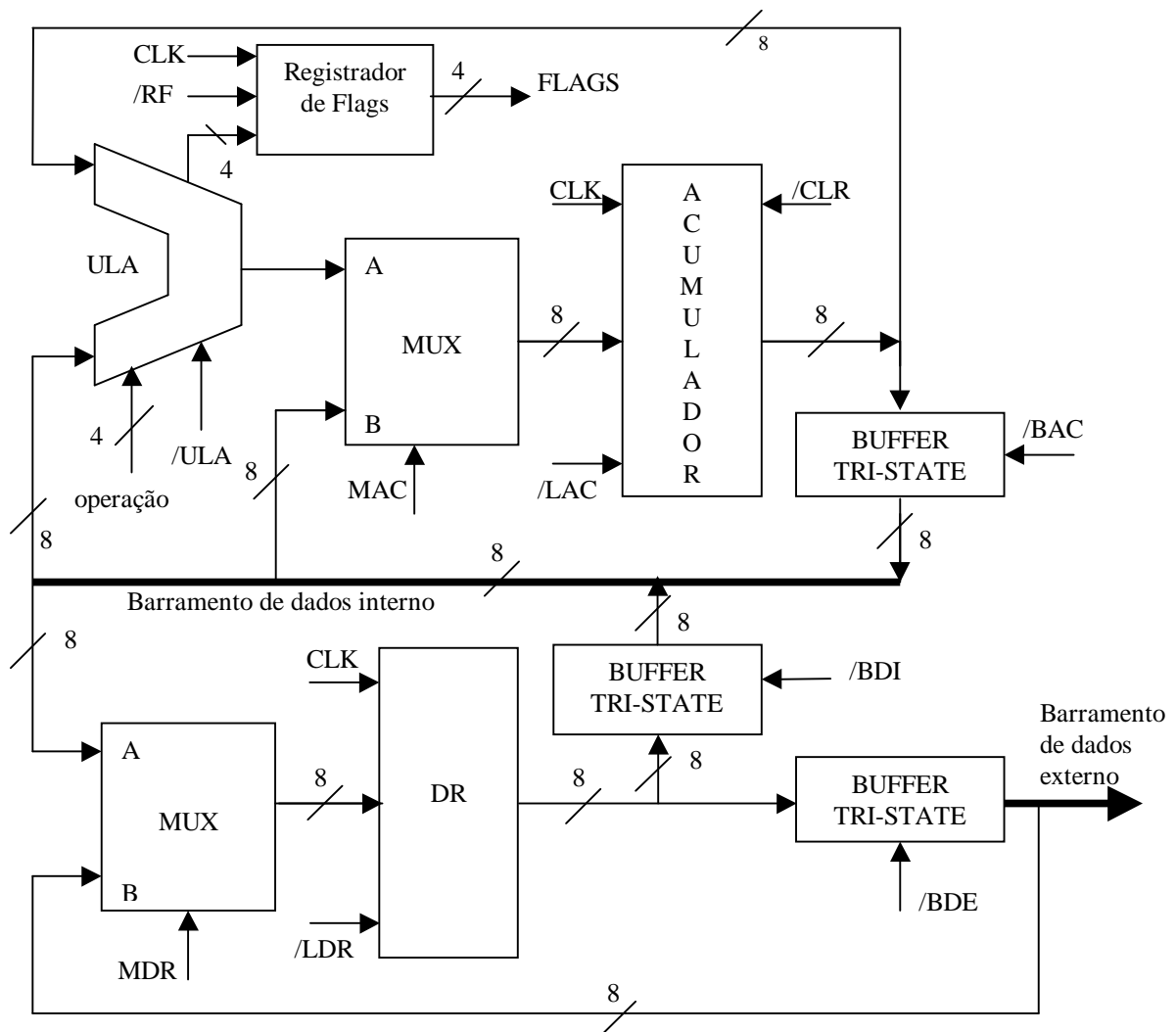
- 8 bits do 1º operando
- 8 bits do 2º operando
- 4 bits do código de operação (vide tabela)
- 1 bit de habilitação da ULA (ativa em zero)

b) Saídas:

- 4 bits de “flags”: carry-out, sinal (positivo, negativo), zero, overflow
- 8 bits do resultado

Se a ULA estiver habilitada, então nos 8 bits do resultado estará o resultado da operação, entre o 1º e 2º operandos, selecionada pelos 4 bits do código de operação.

Os quatro sinais de flags identificam o status da ULA de acordo com o resultado obtido pela realização de uma operação. O flag de zero é ativado (colocado em nível lógico 1) se o resultado possui todos os bits em zero. O flag de sinal é ativado se o resultado da ULA é um número negativo em complemento de dois (bit mais significativo do resultado em 1). O flag de carry-out é ativado se ocorre um vai-um no último bit do resultado. E finalmente o flag de overflow é ativado se o resultado da operação realizada pela ULA não é válido.



**Figura 5:** Diagrama da unidade de dados

Os registradores (acumulador, DR e flags) armazenam o dados temporariamente antes de serem manipulados. O acumulador também é utilizado para o armazenamento do resultado da operação realizada pela ULA. Estes registradores são compostos por flip-flops, onde cada flip-flop é capaz de armazenar um único bit.

Os multiplexadores (MUX) são responsáveis pela seleção de um determinado dado que está em suas duas entradas (A e B). Isto permite a opção de escolha de um dado que será disponibilizado na saída do multiplexador, de acordo com o sinal de controle gerado pela unidade de controle. Por exemplo, o registrador acumulador poderá receber um dado proveniente da saída da ULA ou um dado proveniente do barramento de dados interno.

Os buffers tri-state são simplesmente circuitos que conectam um dos dois dados de entrada na sua saída. Tem como função fazer com que a cada momento apenas um circuito coloque o dado no barramento. Desta forma, o barramento pode ser usado por vários circuitos sem perigo de provocar um curto-circuito.

Os barramentos são simplesmente fios condutores por onde os sinais elétricos dos dados trafegam, sendo um fio para cada bit de dado.

Os sinais de controle da unidade de dados, os quais provêm da unidade de controle, possuem as seguintes funções:

- /ULA – habilita a ULA, ativo em zero
- operação – (4 bits) seleciona a operação que a ULA realiza de acordo com a Tabela 1

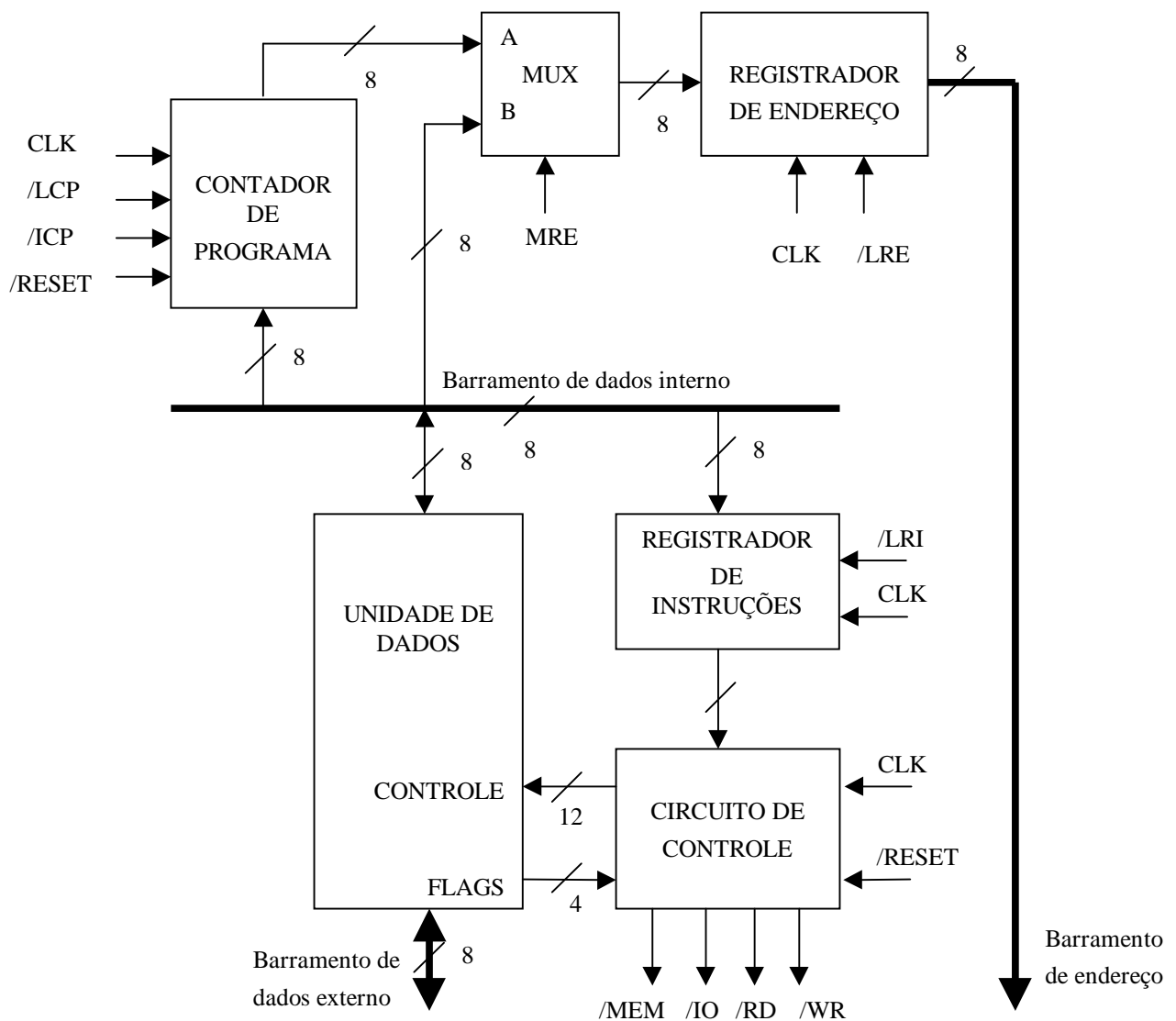
- /RF – habilita a carga do registrador dos flags, ativo em zero, sensível a transição positiva do clock
- MAC – seleciona o dado a ser armazenado no acumulador (0 – A, 1 – B)
- /LAC – habilita a carga do acumulador, ativo em zero, sensível a transição positiva do clock
- /BAC – habilita o buffer do acumulador para envio do dado ao barramento de dados interno, ativo em zero
- MDR – seleciona o dado a ser armazenado no Registrador de Dados (0 – A, 1 – B)
- /LDR – habilita a carga do Registrador de

Dados, sensível a transição positiva do clock

- /BDI – habilita buffer do Registrador de Dados para envio do dado ao barramento de dados interno, ativo em zero
- /BDE – habilita buffer do Registrador de Dados para envio do dado ao barramento de dados externo, ativo em zero

### 3.2 Unidade de controle

A unidade de controle é composta pelo contador de programa, pelos registradores (de endereço e instruções) e circuito de controle. Isto é mostrado na Figura 6.



**Figura 6:** Unidade de controle com a unidade de dados

Os componentes da unidade de controle possuem as seguintes características e funções:

- Registrador de endereço – registrador de 8 bits, com controle de carga (/LRE) e sensível a transição positiva do clock. Tem como função armazenar o endereço da memória que está sendo acessado.
- Registrador de instruções – registrador de 8 bits, com controle de carga (/LRI) e sensível a transição positiva do clock. Tem como função armazenar o código da instrução buscada na memória, e que será executada.
- Contador de programa – contador crescente de 8 bits, com controles de clear assíncrono (/RESET), incremento síncrono (/ICP), e carga síncrona (/LCP), sensível a transição positiva do clock. Tem como função indicar a posição de endereço de memória onde está armazenado a próxima instrução a ser buscada.
- Circuito de controle – máquina sequencial que gera os sinais de controle para a unidade de dados, unidade de controle, porta de entrada/saída e para a memória, de acordo com o dado armazenado no registrador de instruções. Sensível a transição negativa do clock. A máquina sequencial permite que a CPU execute as instruções indicadas na Tabela 2.

Os sinais de controle da unidade de controle são:

- /LCP – habilita a carga do contador de programa, ativo em zero, sensível a transição positiva do clock.
- /ICP – habilita a contagem (incremento) do contador de programa, ativo em zero, sensível a transição positiva do clock.
- /RESET – limpa (zera) o contador de programa e inicializa o circuito de controle, ativo em zero, assíncrono.
- MRE – seleciona o dado a ser armazenado no Registrador de Endereço.
- /LRE – habilita a carga do Registrador de Endereço, ativo em zero, sensível a transição positiva do clock.
- /LRI – habilita a carga do Registrador de Instruções, ativo em zero, sensível a transição positiva do clock.
- FLAGS - (4 bits) identifica o status da ULA

**Tabela 2:** Código de instruções

Código Instrução (hexadecimal)	Descrição	Símbolo
00	Não faz nada	NOP
08	Soma: $ACC=ACC+M(X)$	ADD X
10	Subtração: $ACC=ACC-M(X)$	SUB X
18	$ACC=ACC$ and $M(X)$	AND X
20	$ACC=ACC$ or $M(X)$	ORA X
28	$ACC=ACC$ xor $M(X)$	XOR X
30	Complementa ACC: $ACC=ACC$	NOT
38	$ACC = ACC + 1$	INC
40	$ACC = ACC - 1$	DEC
48	$ACC = ACC/2$	SHR
50	$ACC = ACC*2$	SHL
58	$ACC = -ACC$	NEG
80	$ACC = 0$	CLR
88	$ACC = M(X)$	LDA X
90	$M(X) = ACC$	STA X
A0	Pula incondicionalmente para posição X	JMP X
A8	Pula para posição X se $ACC = 0$	JEZ X
B0	Pula para posição X se ACC positivo ou zero	JGE X
B8	Pula para posição X se ACC negativo	JLZ X
C0	Pula para posição X se houve vai-um	JMC X
C8	Pula para posição X se houve overflow	JMO X
D0	Lê a porta de entrada e armazena no ACC	INP
D8	Escreve o conteúdo de ACC na porta de saída	OUT
FF	Pára o processamento	HLT

X – endereço de memória de 8 bits

M - memória

ACC – Registrador acumulador

## 4 Funcionamento da CPU

A CPU possui o funcionamento descrito a seguir. Inicialmente, o sinal de reset é ativado, fazendo o contador de programa começar com o valor zero. A CPU então está pronta para executar instruções armazenadas na memória, obedecendo dois ciclos de funcionamento: ciclo de busca da instrução (ciclo de fetch), e ciclo de execução

### 4.1 Ciclo de busca da instrução

Após a ativação do sinal de reset, é carregado o valor do contador de programa no barramento de endereço. A memória é ativada para leitura, e seu dado é carregado no registrador de instruções. O contador de programa é incrementado automaticamente preparando para ler o próximo dado na memória.

### 4.2 Ciclo de execução

Após o ciclo de fetch, a instrução carregada é decodificada pelo circuito de controle que gera os

sinais de controle em seqüência para execução da instrução. Se um dado é buscado na memória na execução da instrução, o contador de programa é atualizado. Quando termina de executar a instrução, o ciclo de fetch é reiniciado com o novo valor do contador de programa.

### 4.3 Circuito de controle

O circuito de controle é a parte mais complexa da CPU. Existem várias formas para implementação do circuito de controle [1]: microprogramado, máquina de estado e contador de seqüência. O método proposto é o contador de seqüência por ser mais simples. Ele é mostrado na Figura 7 e descrito a seguir.

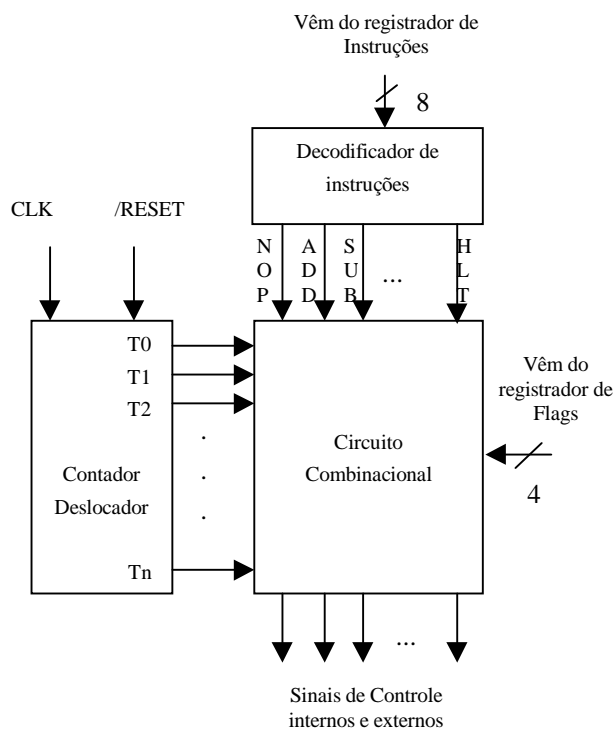


Figura 7: Circuito de controle

O contador deslocador gera sinais sequenciais a cada pulso de clock (transição negativa) de acordo com as formas de onda mostradas na Figura 8. A sua função é marcar o tempo e ordem de ativação dos sinais de controle.

O decodificador de instruções ativa a sua saída correspondente a instrução armazenada no registrador de instruções. O circuito combinacional gera os sinais de controle na seqüência apropriada para busca da instrução (durante ativação de T0, T1 e T2), e posteriormente gera os sinais de controle para

execução da correspondente instrução (durante ativação de T3 à Tn).

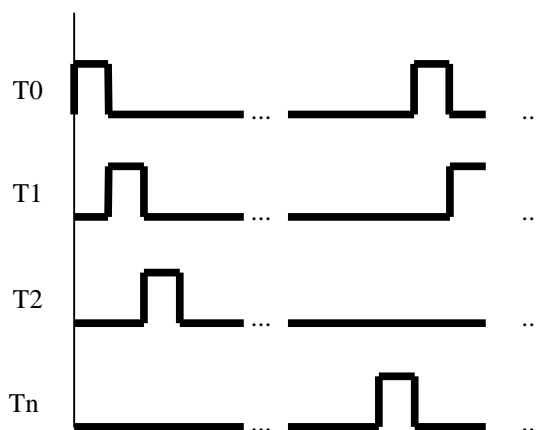


Figura 8: Sinais gerados pelo contador deslocador

### 5 Conclusão

Neste artigo foi apresentada uma proposta de projeto de um processador digital (CPU) simples. O projeto é utilizado na disciplina de Arquitetura de Computadores por ser de fácil assimilação pelos estudantes do curso, além de providenciar um aprendizado eficaz no estudo do funcionamento de um processador. O projeto pode ser implementado utilizando-se circuitos integrados comuns de fácil aquisição, ou mesmo através de programação de dispositivo FPGA. A proposta da CPU pode também ser simulada em computador utilizando-se programas simuladores de circuitos digitais.

### 6 Referências bibliográficas

- [1] HAYES, John Patrick. Computer Architecture and Organization. Singapore: McGraw-Hill International, 2ª edição, 1988, 702p.
- [2] MALVINO, Albert P. Microcomputadores e Microprocessadores. São Paulo: McGraw-Hill do Brasil, 1985, 578p.
- [3] PATTERSON, David A. e HENNESSY, John L. Computer Organization and Design: The Hardware/Software Interface. San Francisco: Morgan Kaufmann, 2ª edição, 1998, 980p.
- [4] TANENBAUM, Andrew S. Organização Estruturada de Computadores. Rio de Janeiro: Livros Técnicos e Científicos, 1999, 3ª edição, 460p.