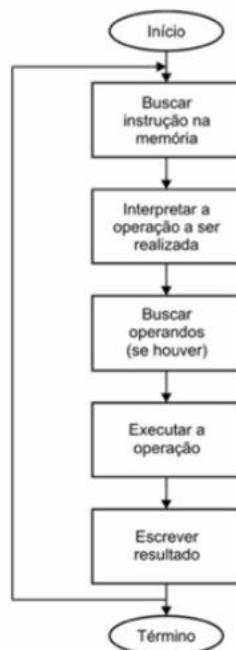


Funcionamento da UCP – O ciclo da instrução

Um programa, para ser efetivamente executado pelo processador deve ser constituído de uma série de instruções de máquina. Para que a execução tenha início, as instruções devem ser armazenadas em células sucessivas, na memória principal.

A função da UCP consiste, então, em:

- buscar uma instrução na memória (operação de leitura), uma de cada vez;
- interpretar que operação a instrução está explicitando (pode ser uma soma de dois números, uma multiplicação, uma operação de entrada ou de saída de dados, ou ainda uma operação de movimentação de um dado de uma célula para outra);
- buscar os dados onde estiverem armazenados, para trazê-los até a UCP;
- executar efetivamente a operação com o(s) dado(s), guardar o resultado (se houver algum) no local definido na instrução; e, finalmente,
- reiniciar o processo apanhando nova instrução.



Estas etapas compõem o que se denomina um ciclo *de instrução*. Este ciclo se repete indefinidamente até que o sistema seja desligado, ou ocorra algum tipo de erro, ou seja encontrada uma instrução de parada. Em outras palavras a UCP é projetada e fabricada com o propósito único de executar sucessivamente pequenas operações matemáticas (ou outras manipulações simples com dados), na ordem e na sequência definidas pela organização do programa.

Base do projeto da UCP → conjunto de instruções que irá executar → operações que o hardware será capaz de realizar diretamente através do circuito.

Depois: definir e especificar os componentes da arquitetura e organização.

Conforme vimos anteriormente, a UCP é responsável por executar os programas, que são compostos de instruções. Agora, veremos como isto é feito, o que chamamos de ciclo de instrução. Cada uma destas instruções é dividida em etapas.

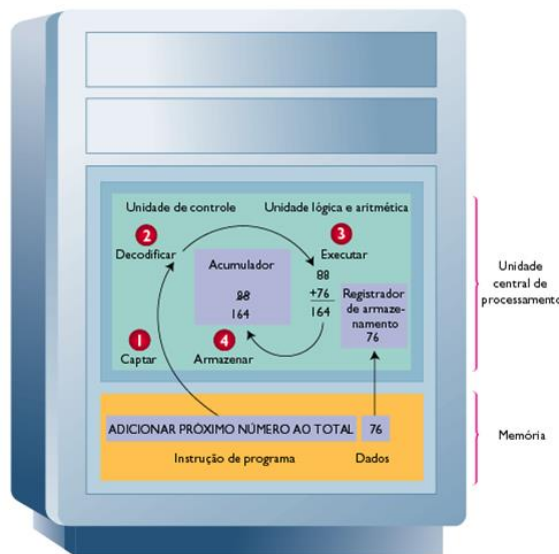
Tanenbaum (2007) define as seguintes etapas, chamadas de ciclo buscar-decodificar-executar: 1º.) Trazer a próxima instrução da memória até o registrador;

- 2º.) Alterar o contador de programa para indicar a próxima instrução;
- 3º.) Determinar o tipo de instrução trazida;
- 4º.) Se a instrução usar uma palavra na memória, determinar onde está a palavra;
- 5º.) Trazer a palavra para dentro de um registrador da CPU, se necessário;
- 6º.) Voltar à etapa 1 para iniciar a execução da instrução seguinte.

Capron (2008) apresenta um exemplo simples do ciclo de instrução:

“Suponhamos que um programa queira encontrar a média de cinco pontuações de exame. Para isso, deve totalizar as cinco pontuações e depois dividir o resultado por cinco. O programa inicia fixando o total em 0. Depois, adiciona cada um dos cinco números, um por vez, ao total. Suponhamos que as pontuações sejam 88, 76, 91, 83 e 87. Nesse cálculo, o total foi fixado em 0 e, depois, 88, a primeira pontuação do exame, foi adicionada. Examine agora o ciclo de máquina quando ele adiciona a pontuação seguinte, 76, ao total. Siga as etapas do ciclo de máquina:

- (1) Captar: a unidade de controle capta a instrução ADD da memória.
- (2) Decodificar: a unidade de controle decodifica a instrução ADD. Ela determina que a adição deve ser efetuada e fornece instruções para que o número seguinte, 76, seja colocado em um registrador para esse propósito. O total atual, 88, já está no acumulador.
- (3) Executar: o número seguinte, 76, é colocado no registrador. A ALU faz a adição, aumentando o valor para 164.
- (4) Armazenar: nesse caso, a ALU armazena o resultado no acumulador, em vez de na memória, porque ainda é necessário adicionar mais números a ela. O novo total, 164, substitui o total anterior, 88 e o processo continua”.



O ciclo de máquina em ação

Fonte: Capron (2008).

Para descrever as etapas de um ciclo de instrução será utilizada uma organização de UCP/MP hipotética, mais simples.

Características de um processador simples

- 1) Palavra: 12 bits
- 2) Endereços: 8 bits (256 células de memória)
- 3) Células de 12 bits
- 4) Instruções de 1 operando apenas, com C. Op. = 4 bits e campo operando = 8 bits

- 5) O campo operando sempre indica o endereço de memória do dado, exceto em instruções de desvio
- 6) A UCP possui apenas um registrador de dados, o ACC, com 12 bits de tamanho, o RI, também com 12 bits de tamanho, o CI e o REM, com 8 bits cada um, e o RDM, com 12 bits também.

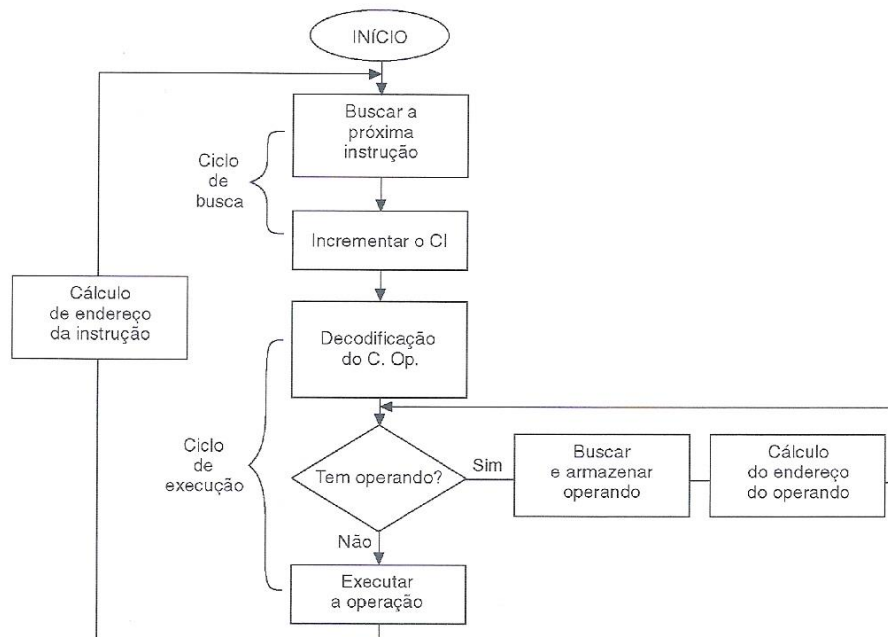
7) Formato das instruções

C. Op.	Operando
4 bits	8 bits

8) Instruções disponíveis

C. Op.	Sigla	Descrição
0	HLT	Parar a execução do programa
1	LDA Op	$ACC \leftarrow (Op)$
2	STR Op	$(Op) \leftarrow ACC$
3	ADD Op	$ACC \leftarrow ACC + (Op)$
4	Sub Op	$ACC \leftarrow ACC - (Op)$
5	JZ Op	Se $ACC = 0$, então: $CI \leftarrow Op$
6	JP Op	Se $ACC > 0$, então: $CI \leftarrow Op$
7	JN Op	Se $ACC < 0$, então: $CI \leftarrow Op$
8	JMP Op	$CI \leftarrow Op$
9	GET Op	Ler dados da porta de entrada e armazená-lo em (Op)
A	PRT Op	Colocar na porta referente à impressora o valor armazenado em (Op)

A figura a seguir mostra o fluxo de ações que constituem um ciclo de instrução de uma forma mais detalhada.



Em linguagem de transferência entre registradores pode-se escrever:

- Iniciar
- $RI \leftarrow (CI)$
- $(CI) \leftarrow (CI) + 1$

- interpretar o código de operação
- buscar operando
- executar a instrução
- retornar

Obs.: não será considerada interrupção nem pipeline.

Execução da instrução LDA Op

- Valores iniciais:

a) LDA → armazenada no endereço 02H

Conteúdo: 1B4

0011	10110100
Cód. Op.	Operando

b) dado do endereço B4 → 1A7

c) ADD → armazenada no endereço 03H

Conteúdo: 3B5

0011	10110101
Cód. Op.	Operando

d) valor em B5 → 07D

e) valor no CI → 02 H

f) valor em RI = 317 (valor da instrução anterior)

g) valor no Acc ou registrador de dados = 20B (valor de operação anterior)

h) valor no REM – B3

i) valor no RDM = 7BC

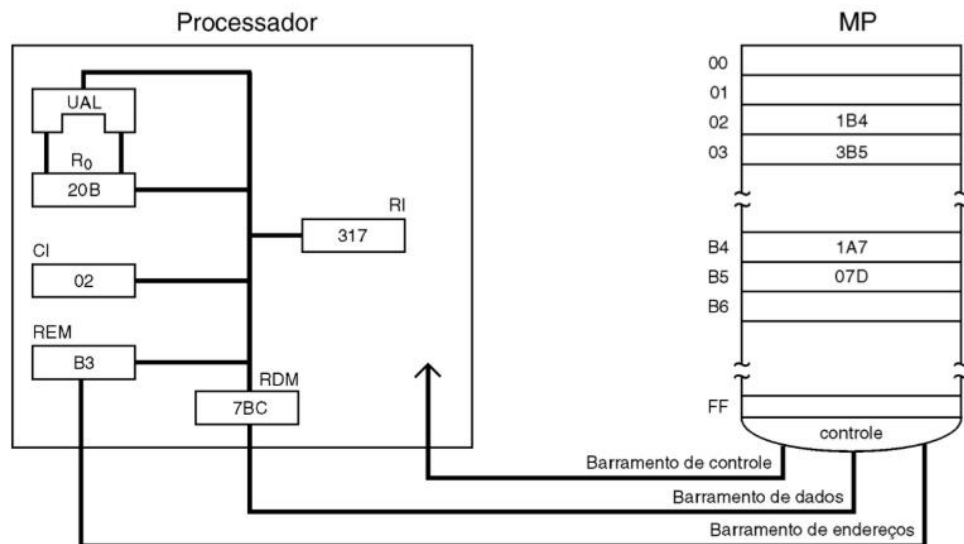


Figura 6.25 Dados iniciais para execução do ciclo das instruções LDA e ADD.

- Foi terminado o ciclo de uma instrução.
A UC reinicia o processo (ciclo de uma nova instrução → LDA)
De acordo com o fluxograma:
- 1) $RI \leftarrow (CI)$
 - UC aciona a transferência para o REM pelo barramento

REM = CI = 2

- UC ativa a linha READ (leitura) do barramento de controle
- Decodificado o endereço, o controle da MP transfere os bits (da célula 02) para o RDM (pelo barramento de dados)

RDM = 1B4

- O valor é transferido para o RI (pelo barramento interno)

2) $(CI) \leftarrow (CI) + 1$

CI = 03

Mas, geralmente, a instrução ocupa mais de uma célula, o que resulta:

$(CI) \leftarrow (CI) + n$, onde n é a quantidade de células ocupadas por uma instrução.

Os 2 primeiros bits do código de operação podem indicar o tamanho da instrução quando n é variável.

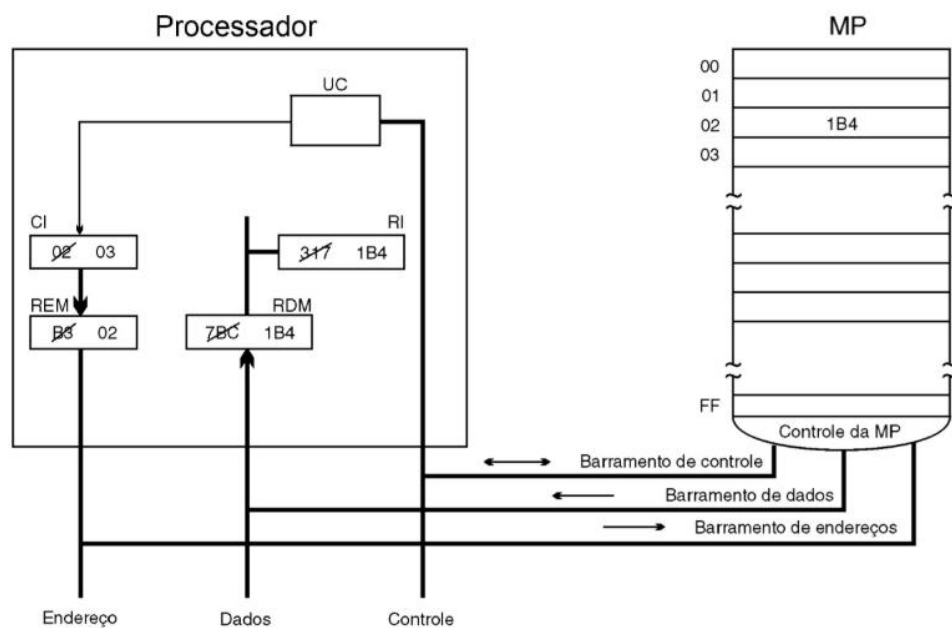


Figura 6.26 Fluxo de dados e de endereços durante a realização do ciclo de busca da instrução LDA.

- 3) Decodificação do código de operação
 - a) UC emite sinal para que o RI transfira para o decodificador de instrução os 4 bits mais significativos (código de operação)
 - b) O decodificador seleciona (através de sua lógica) a linha de saída correspondente para a UC, que emitirá sinais adequados e em sequência, para a execução da operação.
- 4) Se tiver operandos, buscá-los
Não há operando.
- 5) Execução da operação
Acc = 1A7
 - a) a UC emite os sinais para que os bits do valor do campo do operando (B4) sejam transferidos para o REM (pelo barramento interno)
 - b) A UC ativa a linha READ (leitura) do barramento de controle → aciona o controle da MP para decodificar o endereço B4
 - c) O controle da MP transfere o conteúdo da célula de endereço B4 (1A7) para o RDM (pelo barramento de dados)

RDM = 1A7

- d) a UC emite o sinal para que o valor seja transferido para o Acc (pelo barramento interno).

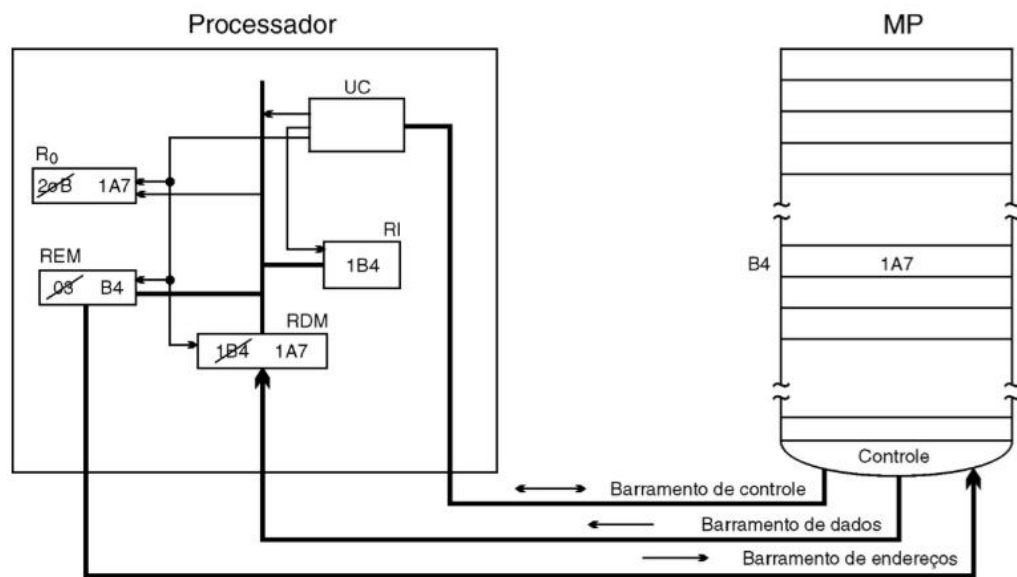


Figura 6.27 Fluxo de dados e de endereços durante o ciclo de execução da instrução LDA.

1 e 2 → Ciclo de busca

3, 4 e 5 → Ciclo de execução