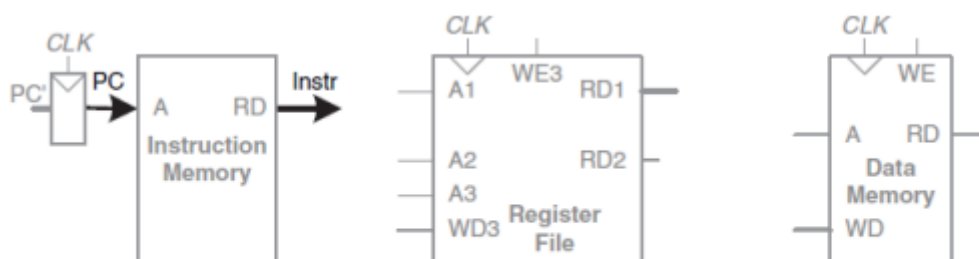


# PROCESSADOR CICLO ÚNICO

## Caminho de Dados de Ciclo-único

- PÁGINA 563

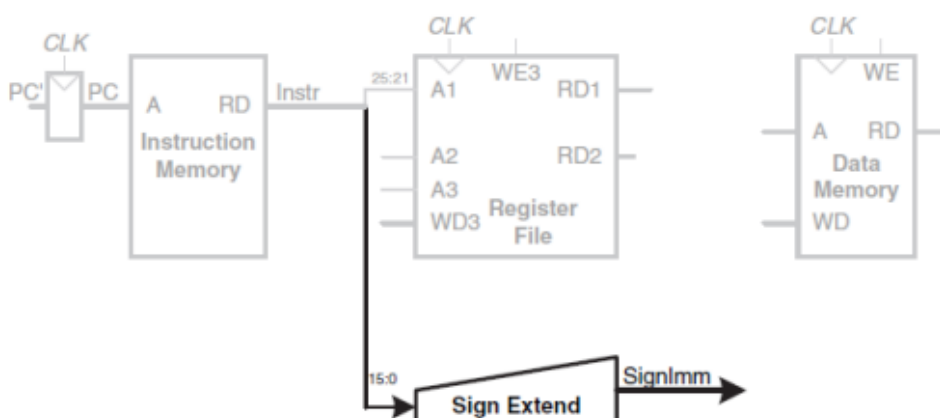
O registro **contador de programa (PC)** contém o endereço da instrução a executar, O primeiro passo é ler esta instrução da **memória de instruções**. A **memória de instruções** lê, ou fetches, a instrução de 32-bits, identificada como **Instr**.



**Figura 7.2** Leitura (*fetch*) de instrução da memória

O próximo passo é ler o registro de origem que contém o endereço de base. Este registro é especificado no campo rs da instrução, **Instr<sub>25:21</sub>**. Estes bits da instrução estão ligados à entrada de endereço de um dos portos de leitura do banco de registros, A1. O banco de registros lê o valor de registro em **RD1**.

O **offset** é armazenado no campo imediato da instrução, **Instr<sub>15:0</sub>**. Uma vez que o 16-bit imediato pode ser positivo ou negativo, ele deve ter o sinal estendido para 32 bits. O valor de 32 bits de sinal estendido é chamado **SignImm**



**Figura 7.4** Extensão de sinal imediato

O processador deve adicionar o endereço de base ao offset para encontrar o endereço a ler a partir da memória -> **ALU** faz a adição.

A **ALU** recebe dois operandos, **SrcA** e **SrcB**. O **SrcA** vem do banco de registros, e o **SrcB** vem do imediato com extensão de sinal.

O sinal de 3-bits de controlo da **ALU** especifica a operação. A **ALU** gera um **ALUResult** de 32-bits e uma **flag Zero**, que indica se **ALUResult** == 0. O **ALUResult** é enviado para a memória de dados (**Data Memory**) como o endereço da instrução de carregamento.

Os dados são lidos a partir da memória de, em seguida, é escrita de volta para o registro de destino (**ReadData**) no banco de registros no final do ciclo, o Porto 3 do banco de registros é o porto de escrita. O registro de destino para a instrução lw é especificado no campo rt, **Instr<sub>20:16</sub>**, o qual está ligado à entrada de endereço do porto 3, A3, do banco de registros. Em seguida, o barramento de leitura de dados **ReadData** está ligado à entrada de escrita de dados do porto 3, WD3, do banco de registros. Um sinal de controle chamado de **RegWrite** está ligado à entrada de **enable** de escrita do porto 3, **WE3**, e é ativado durante uma instrução lw para que o valor do dado seja escrito no banco de registros.

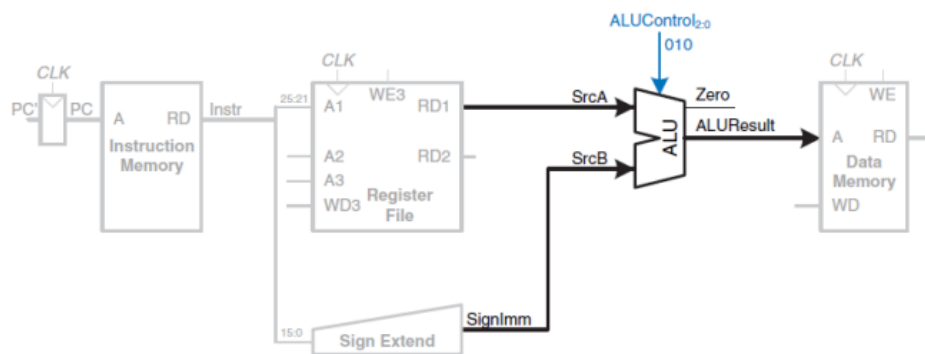


Figura 7.5 Endereçamento de memória do computador

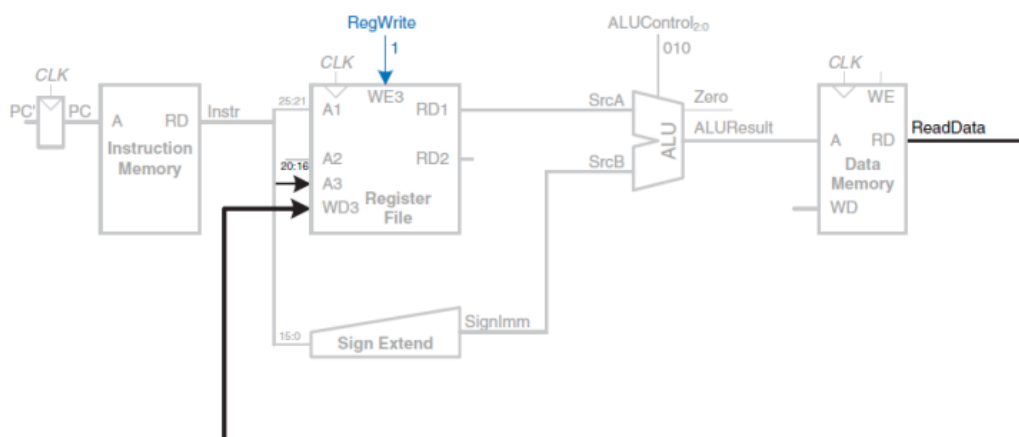


Figura 7.6 Escrita de retorno dos dados no registro

Enquanto a instrução está a ser executada, o processador deve calcular o endereço da próxima instrução, PC'. Porque as instruções têm 32 bits = 4 bytes, a próxima instrução está em PC + 4.

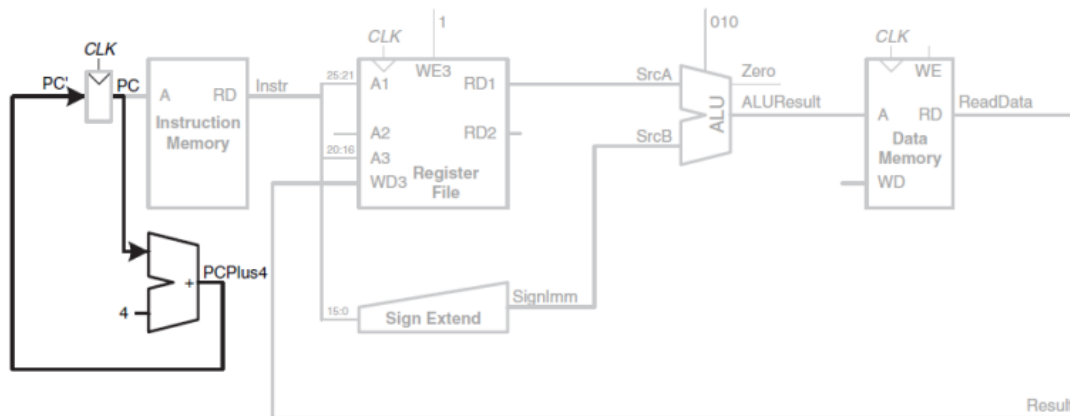


Figura 7.7 Determinação do endereço da próxima instrução para o PC

Consider **subset** of ARM instructions:

– **Data-processing instructions:**

- **ADD, SUB, AND, ORR**
- with register and immediate Src2, but **no shifts**

– **Memory instructions:**

- **LDR, STR**
- with **positive immediate offset**

– **Branch instructions:**

- **B**

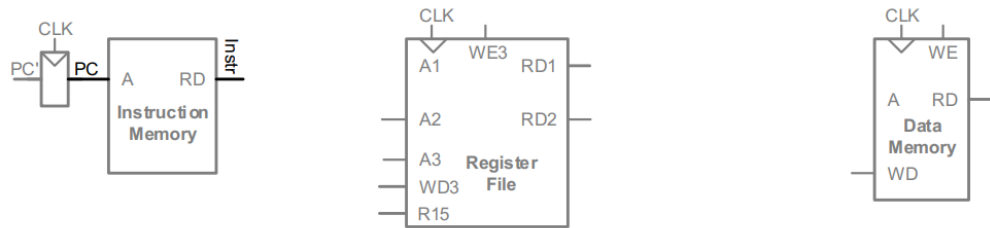
**Determines everything about a processor:**

– Architectural state:

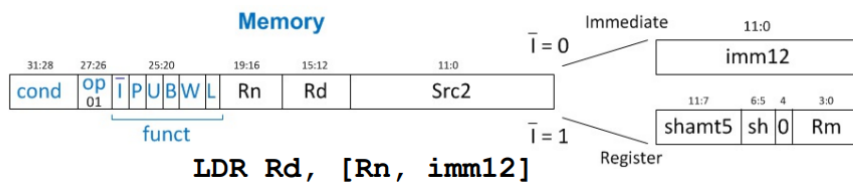
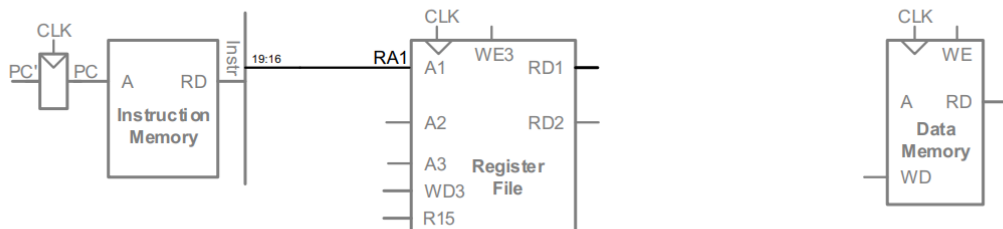
- 16 registers (including PC)
- Status register

– Memory

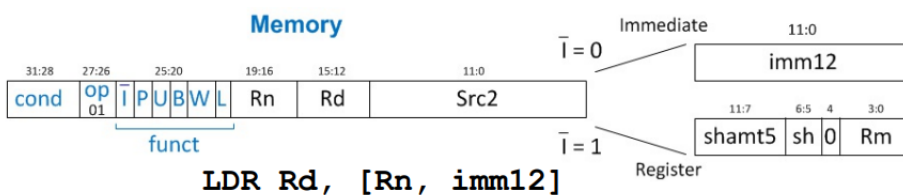
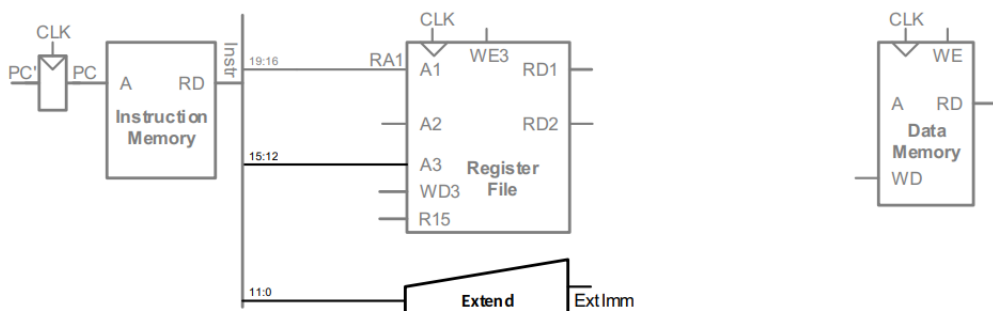
## STEP 1: Fetch instruction



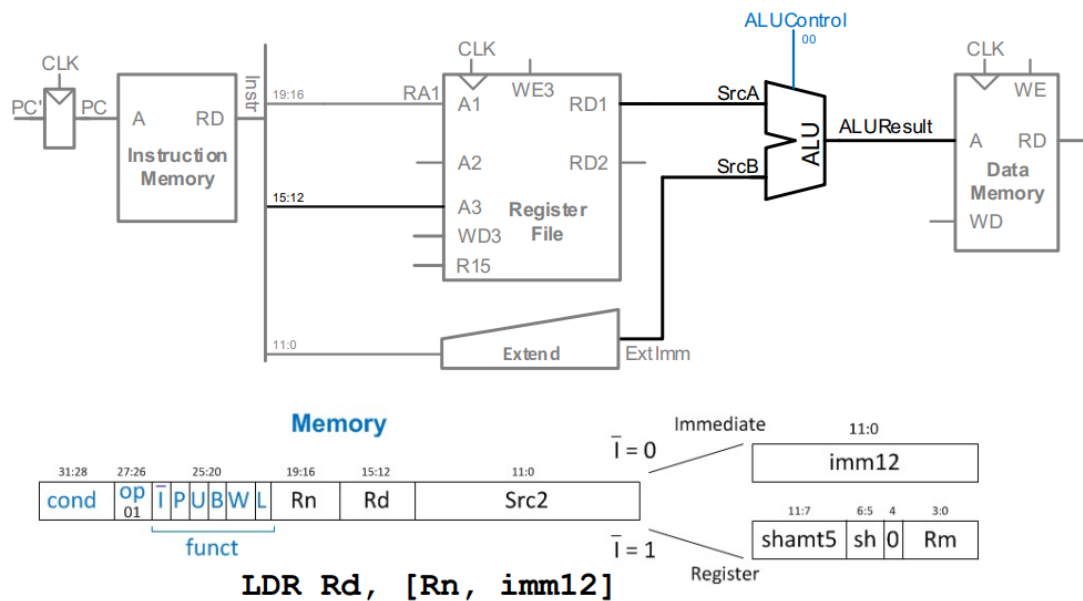
## STEP 2: Read source operands from RF



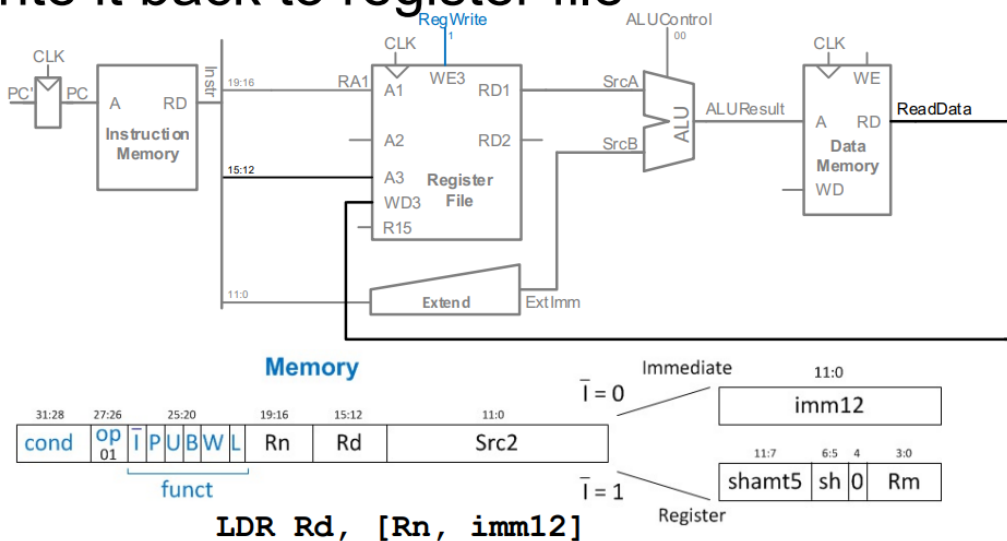
## STEP 3: Extend the immediate



### STEP 4: Compute the memory address



**STEP 5:** Read data from memory and write it back to register file



### STEP 6: Determine address of next instruction

