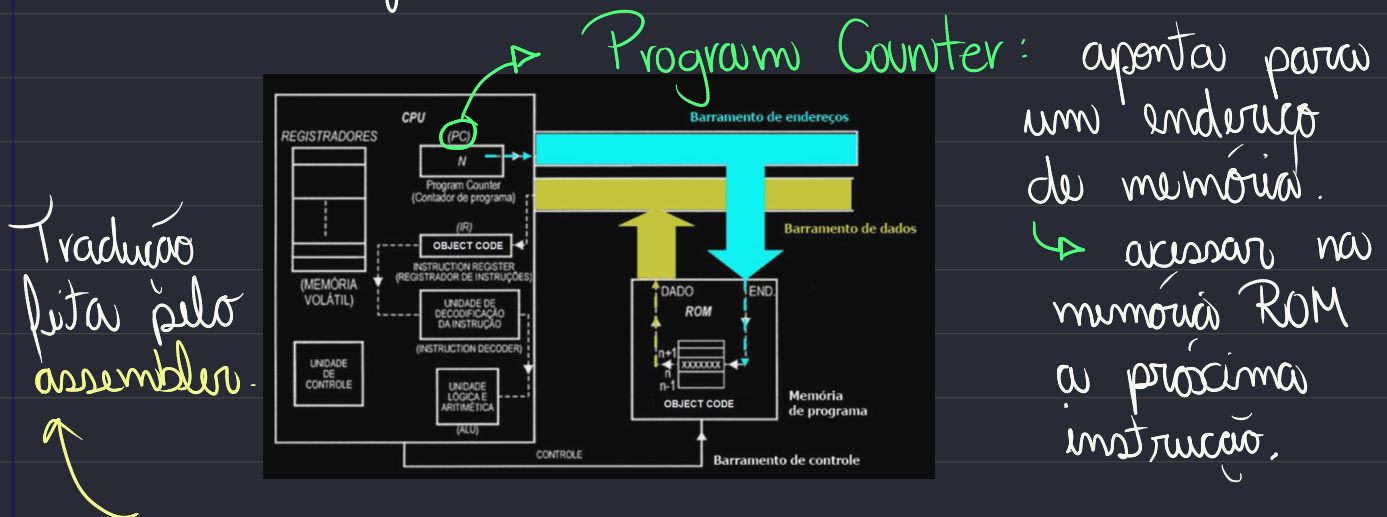


# LINGUAGEM ASSEMBLY

- ⊛ É quase uma linguagem de máquina, mas ainda é legível para humanos, portanto é apenas uma linguagem de baixo nível.



- ⊛ Assembly e outras linguagens são traduzidas para linguagem de máquina.

- ⊛ É uma linguagem de máquina com **mnemônicos**.

⚡ O assembler é responsável por fazer essa conversão. **ADD, MOV, SUB**

⇒ Thumb-2

- Para executar algumas instruções, é necessário escolher entre 16 e 32 bits. O thumb-2 facilita esse processo, alternando automaticamente entre os dois.

- ⊛ Load / store.   
 → sempre que acessar a memória.   
 → somente memória RAM.

## ⇒ Instruções básicas Assembly.

Por exemplo, uma função que recebe R0 como entrada:

1	Func	MOV	R1	#100	; R1=100
2		MUL	R0,	R0, R1	; R0=100*input
3		ADD	R0,	#10	; R0=100*input+10
4		BX	LR		; retorna 100*input+10

imediato

⊗ Hexadecimal      0x número.  
 Binário              2\_ número.

## ⇒ Algumas comandos

load → LDR → Desloca bits para a direita.  
 ↳ LDR → Pode acessar a memória.  
 store → STR → Salva na memória.

Load

LDR (R0), [R1]  
 ↳ destino recebe endereço de R1  
 ↳ passa seu endereço para o registrador de destino.

store

STR (R2), [R0, #4]  
 ↳ armazena o seu valor no destino.  
 ↳ salva o valor de R2 em R0 + #4 endereço.

## ⇒ Tipos de Acesso à Memória

Modo	Mnemônico Assembly	Endereço Acessado	Valor Final no Registrador Base
Com Offset, base não alterada	LDR R0, [R1, #d]	$R1 + d$	R1
Pré-indexado, base alterada	LDR R0, [R1, #d]!	$R1 + d$	$R1 + d$
Pós-indexado, base alterada	LDR R0, [R1], #d	R1	$R1 + d$

## ⇒ Operações de Transfêrência

MOV ⇒ Menos significativos.  
MOVT ⇒ Mais significativos.

⇒ Ao utilizar esse comando, os valores mais significativos serão zerados como 0.

## ⇒ Opções de Deslocamento

- deslocamento lógico: desloca e insere zeros.
- deslocamento aritmético: desloca mas mantém o último.
  - ⇒ útil para números com sinal.
- rotação: ao invés do bit sair, ele retorna pelo outro lado, portanto não há substituição por zeros ou uns.
- rotação estendida: antes de ser rotacionado o bit vai para a flag.
  - ⇒ um pouquinho apenas.

## ⇒ Operações Aritméticas

- Flags {
- soma: soma coube ou não coube nos 32 bits.
  - subtração: resultado neg. ou positivo.
  - overflow: alguma coisa estranha aconteceu.

## ⇒ Pilhas

PUSH <reglist> ⇒ Enviar apenas um por  
POP <reglist> reg.