Microprocessadores

Arquitecturas Aritméticas
Controladores

António M. Gonçalves Pinheiro

Departamento de Física
Universidade da Beira Interior
Covilhã - Portugal

pinheiro@ubi.pt

Aritmética Binária - Números Inteiros

Números Inteiros sem sinal

Usam normalmente a representação binária.

com um byte: $b_7b_6b_5b_4b_3b_2b_1b_0$ entre 0 e 2^8-1=255
com 16 bits: $b_{15}b_{14}....b_3b_2b_1b_0$ entre 0 e 2^{16}-1
com N bits: $b_{N-1}b_{N-2}....b_3b_2b_1b_0$ entre 0 e 2^N-1

Aritmética Binária - Números Inteiros

Soma de Números Inteiros sem sinal

$$
\begin{array}{r} 0\;1\;0\;1\;1\;1\;0\;1 \\ +\;1\;0\;0\;1\;1\;1\;1\;0 \\ \hline 1\;1\;1\;1\;1\;0\;1\;1 \end{array}
\qquad
\begin{array}{r} 93 \\ +\;158 \\ \hline 251 \end{array}
$$

Aritmética Binária - Números Inteiros

Soma de Números Inteiros sem sinal

$$
\begin{array}{ccccccccc}
 & & & \color{blue}{\mathbf{1}} & \color{blue}{\mathbf{1}} & \color{blue}{\mathbf{1}} & & & \\
 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
+ & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
\hline
 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1
\end{array}
\qquad
\begin{array}{cccc}
 & & 9 & 3 \\
+ & 1 & 5 & 8 \\
\hline
 & 2 & 5 & 1
\end{array}
$$

Aritmética Binária - Números Inteiros

Soma de Números Inteiros sem sinal - “Overflow

1			**1**	**1**	**1**								
	1	1	0	1	1	1	0	1			2	2	1
+	1	0	0	1	1	1	1	0		+	1	5	8
1	0	1	1	1	1	0	1	1			1	2	3
											3	**7**	**9**

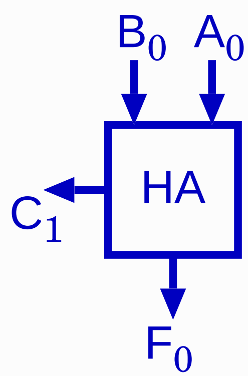
NOTA: Quando o número de bits não é suficiente para representar o resultado final, diz-se que ocorreu um “*overflow*".

Aritmética Binária - Números Inteiros

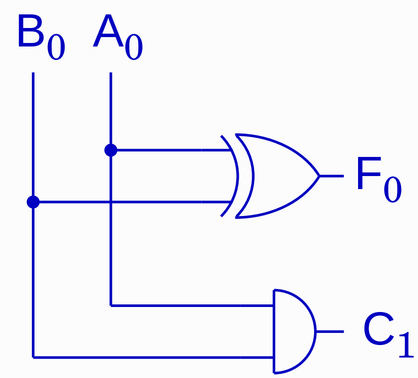
Soma de Números Inteiros sem sinal - Circuitos Somadores

“half-adder

Somador de dois bits



B_0	A_0	C_1	F_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

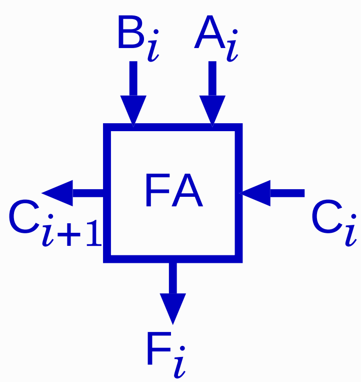


Aritmética Binária - Números Inteiros

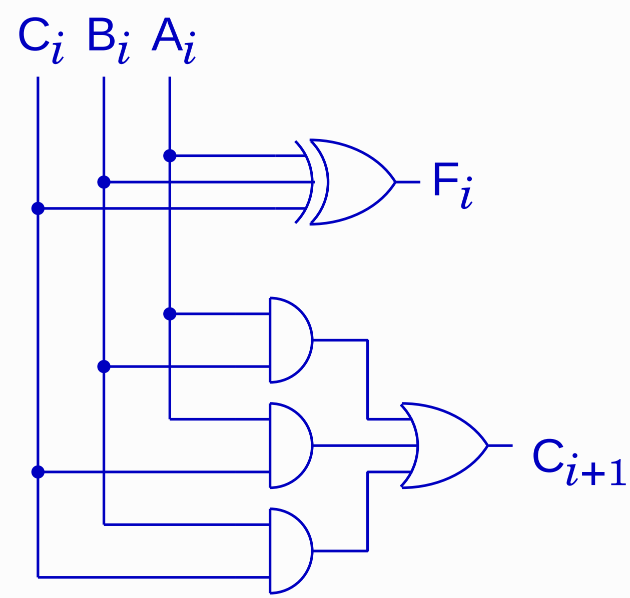
Soma de Números Inteiros sem sinal - Circuitos Somadores

“full-adder

Somador de três bits

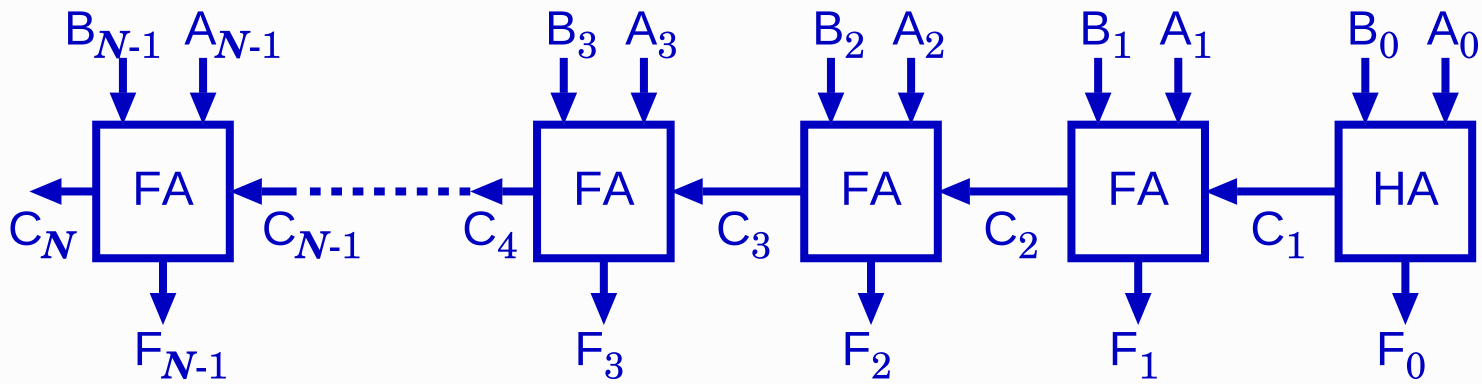


C_i	B_i	A_i	C_{i+1}	F_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Aritmética Binária - Números Inteiros

Somador de Números Inteiros sem sinal com N bits



Aritmética Binária - Números Inteiros

Exemplo

Projecte um circuito baseado numa célula que iterativamente verifique qual é o maior de dois números.

Aritmética Binária - Números Inteiros

Números Inteiros com sinal

Sinal módulo

bit mais significativo representa o sinal ($- \rightarrow 1$)

Ex: 10010111= $-2^4 + 2^2 + 2 + 1 \ = \ -23$

Intervalo Representado com N bits: $\left\{-\left(2^{N-1}-1\right), ..., 2^{N-1}-1\right\}$

Complemento para UM

bit mais significativo representa o sinal ($- \rightarrow 1$)

o módulo de um número negativo obtem-se por negação de todos os bits

Ex: 11101000= - (00010111)= $-(2^4 + 2^2 + 2 + 1) \ = \ -23$

Intervalo Representado com N bits: $\left\{-\left(2^{N-1}-1\right), ..., 2^{N-1}-1\right\}$

Aritmética Binária - Números Inteiros

Números Inteiros com sinal

Complemento para DOIS

bit mais significativo representa o sinal (- $\rightarrow$ 1)

o módulo de um número negativo obtem-se por negação de todos os bits seguido de soma por 1

Ex: 11101001= - (00010110)+1 = - (00010111) = - $2^4 + 2^2 + 2 + 1$ = -23

Intervalo Representado com N bits: $\left\{-\left(2^{N-1}\right), ..., 2^{N-1} - 1\right\}$

NOTA: Mais utilizado pois permite operações aritméticas directas.

100000 em C$_2$ representa o número -2^{N-1}

(Se N=8, 10000000=-128)

Aritmética Binária - Números Inteiros

Soma de Números Inteiros com sinal

$$
\begin{array}{cccccccccccccc}
 & & & \mathbf{1} & \mathbf{1} & \mathbf{1} & & & & & & & & \\
 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & & & & 9 & 3 \\
+ & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & & + & - & 9 & 8 \\
\hline
 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & & & - & & 5
\end{array}
$$

$$1\,0\,0\,1\,1\,1\,1\,0 \xrightarrow{C_2} 0\,1\,1\,0\,0\,0\,1\,0 = 98$$

$$1\,1\,1\,1\,1\,0\,1\,1 \xrightarrow{C_2} 0\,0\,0\,0\,0\,1\,0\,1 = 5$$

Aritmética Binária - Números Inteiros

Soma de Números Inteiros com sinal

$$
\begin{array}{ccccccccccccc}
 & & & \mathbf{1} & \mathbf{1} & \mathbf{1} & & & & & & & \\
 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & & - & 3 & 5 \\
+ & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & + & - & 9 & 8 \\
\hline
 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & & 1 & 2 & 3 \\
 & & & & & & & & & - & \mathbf{1} & \mathbf{3} & \mathbf{3}
\end{array}
$$

$$1\,1\,0\,1\,1\,1\,0\,1 \xrightarrow{C_2} 0\,0\,1\,0\,0\,0\,1\,1 = 35$$

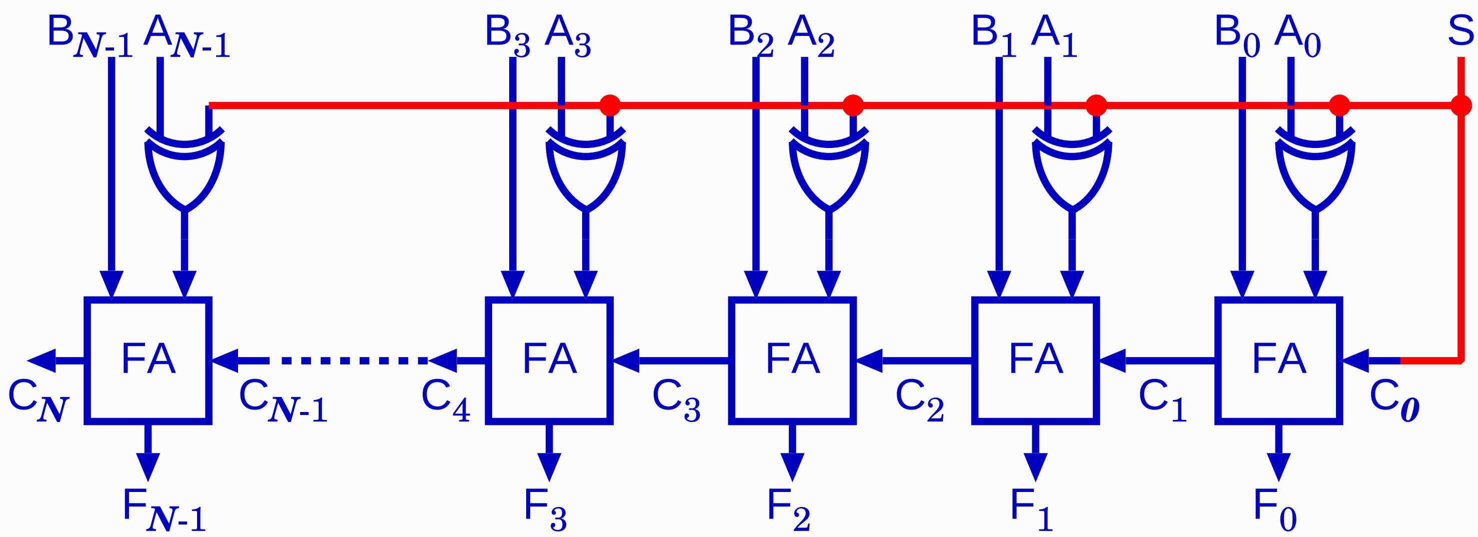
$$1\,0\,0\,1\,1\,1\,1\,0 \xrightarrow{C_2} 0\,1\,1\,0\,0\,0\,1\,0 = 98$$

NOTA: “Overflow"negativo porque $-133 < -128 = -2^7$

Aritmética Binária - Números Inteiros

Somador/Subtractor de Números Inteiros com Sinal com N bits

S=0 F=B+A Somador
S=1 F=B$-$A Subtractor



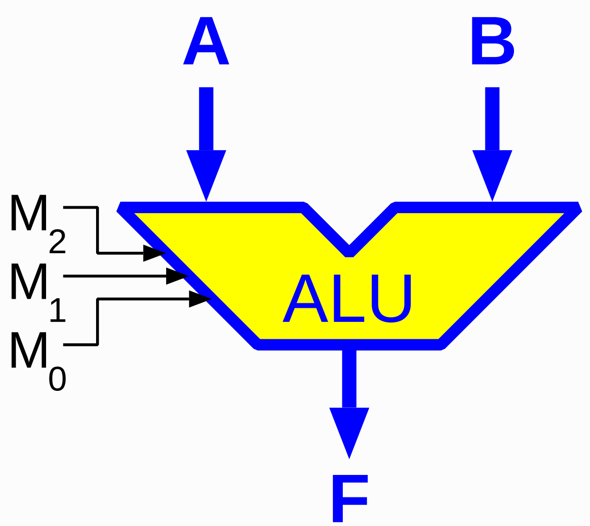
S=1 faz-se negação de todos os bits do número A e soma-se 1 $\rightarrow$ ***Complemento para 2***.

NOTA: $0 \oplus A = A$, $1 \oplus A = \overline{A}$

Aritmética Binária - Números Inteiros

Unidade Lógica e Aritmética - ALU

Sistema combinacional concebido para fazer diferentes operações Aritméticas e Lógicas entre duas palavras binárias



M_2	M_1	M_0	Operação
0	0	0	A + B
0	0	1	A - B
0	1	0	A + 1
0	1	1	A - 1
1	0	0	$A \cdot B$
1	0	1	$A + B$
1	1	0	$A \oplus B$
1	1	1	$\overline{A}$

Nota:
M_2=0 - Operações Aritméticas
M_2=1 - Operações Lógicas

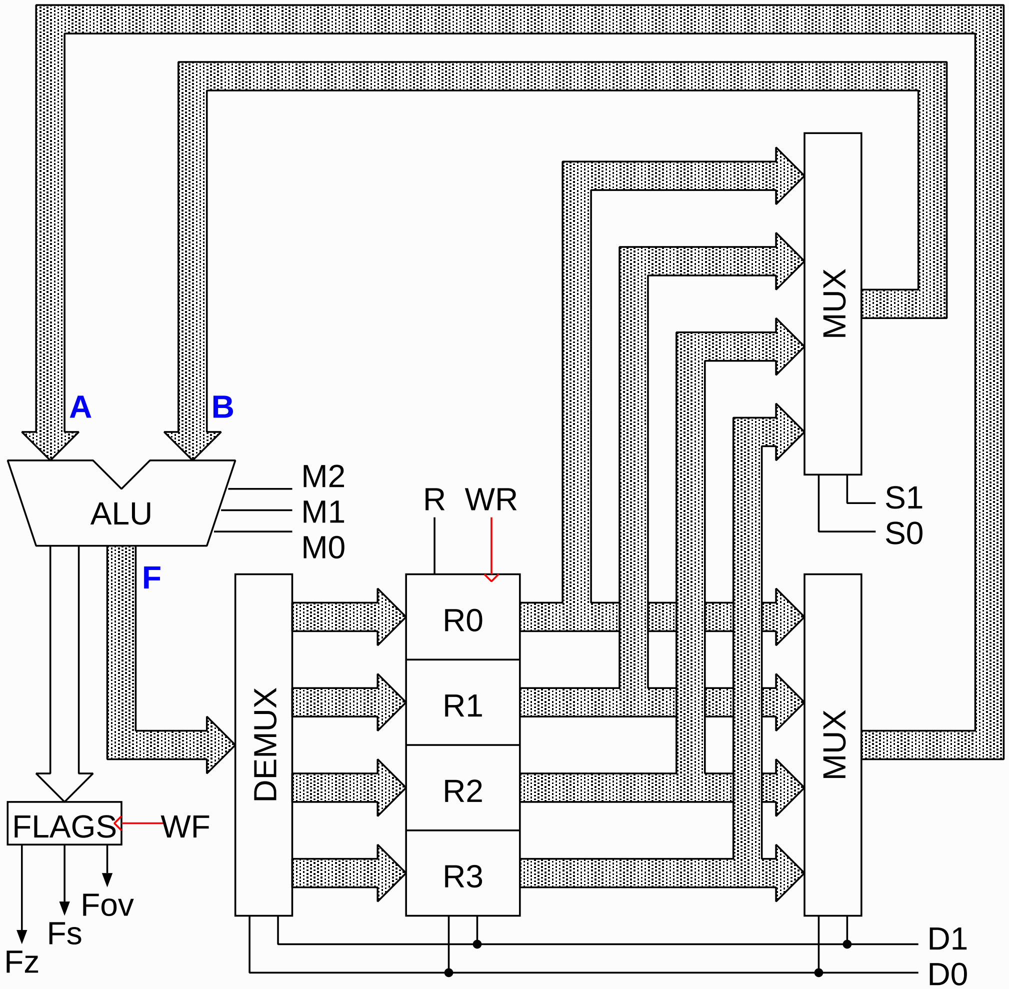
Aritmética Binária - Números Inteiros

Exemplo

Projecte um circuito de que resulte uma variável O_v que para o circuito somador/subtractor estudado sinalize sempre que há um “overflow" negativo ou positivo.

Aritmética Binária - Números Inteiros

Arquitectura Aritmética



M_2	M_1	M_0	Operação
0	0	0	A + B
0	0	1	A - B
0	1	0	A + 1
0	1	1	A - 1
1	0	0	$A \cdot B$
1	0	1	$A + B$
1	1	0	$A \oplus B$
1	1	1	$\overline{A}$

Nota:

M_2=0 - Operações Aritméticas

M_2=1 - Operações Lógicas

Aritmética Binária - Números Reais

Ponto Fixo

$\boxed{} \bullet \boxed{}$

Parte Inteira | Parte Decimal

Ponto Flutuante

$\boxed{}\,1\bullet\boxed{} \times (10)_2^{\boxed{}}$

Sinal | Mantissa | Expoente

Underflow -
Número com valor absoluto muito pequeno diferente de zero que não pode ser representado.
Overflow -
Número com valor absoluto muito grande que não pode ser representado.

FPU - “Floating Point Unit"
Circuito que se destina a fazer operações com números em vírgula flutuante.

Aritmética Binária - Números Reais

Norma do IEEE 754-1985

Nome	Número de bits	Sinal	Mantissa	Expoente	Polarização
Precisão Simples	32	1	23	8	127
Precisão Dupla	64	1	52	11	1023

Valor representado em decimal:

$$(-1)^{\text{Sinal}} \times \left(1 + \sum_{n=1}^{p-1} \text{Mantissa}(n) \times 2^{-n}\right) \times 2^{\text{Expoente-Polarização}}$$

Formato:

Sinal	Expoente	Mantissa

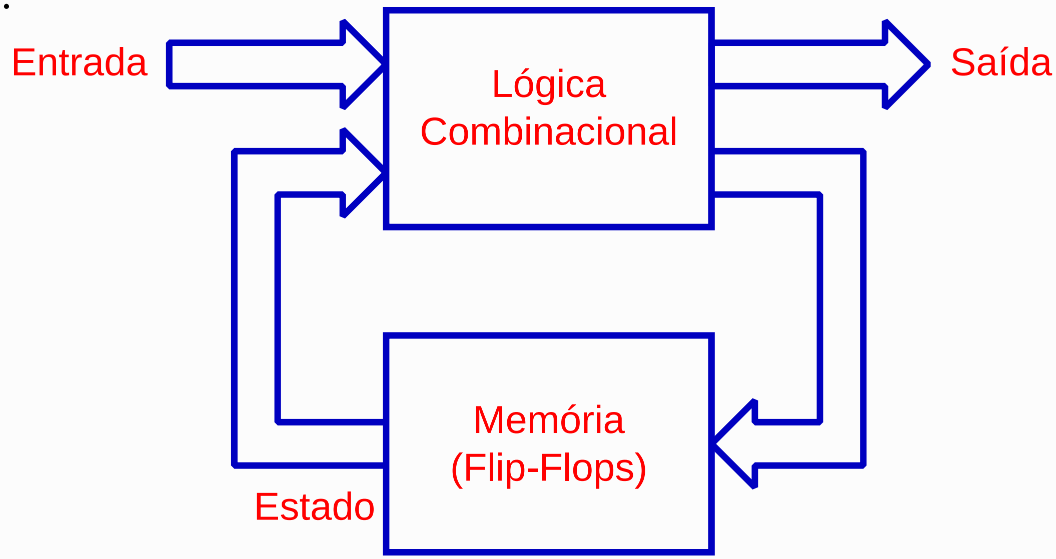
Casos especiais:

Tipo	**Expoente**	**Mantissa**
Zero	00...000	00...000
Infinito	11...111	00...000
NaN	11...111	$\neq 0$
Num. não normalizado	00...000	$\neq 0$

Controladores Digitais

Controladores Digitais:
Circuitos digitais sequenciais síncronos que estabelecem sequências temporais de acordo com entradas de controlo.

Controlador Genérico:



Neste curso são estudados os controladores sequenciais com um **Flip-Flop por estado**.
Este tipo de controladores têm como principal vantagem, a grande simplicidade de projecto.

Controladores Digitais

Exemplo de Controladores Digital com um Flip-Flop por estado:

Projecte um circuito que estabeleça a seguinte sequência de controlo num sistema de luzes com uma lâmpada Vermelha, Azul e Verde:

O sistema tem uma variável **M** que controla a sequência. da seguinte forma:

Se **M**=0 $\Longrightarrow$ Vermelho $\longrightarrow$ Verde+Azul $\longrightarrow$ (Tudo apagado) $\longrightarrow$ (Volta ao princípio)

Se **M**=1 $\Longrightarrow$ Vermelho $\longrightarrow$ Verde $\longrightarrow$ Azul $\longrightarrow$ (Volta ao princípio)

Controladores Digitais

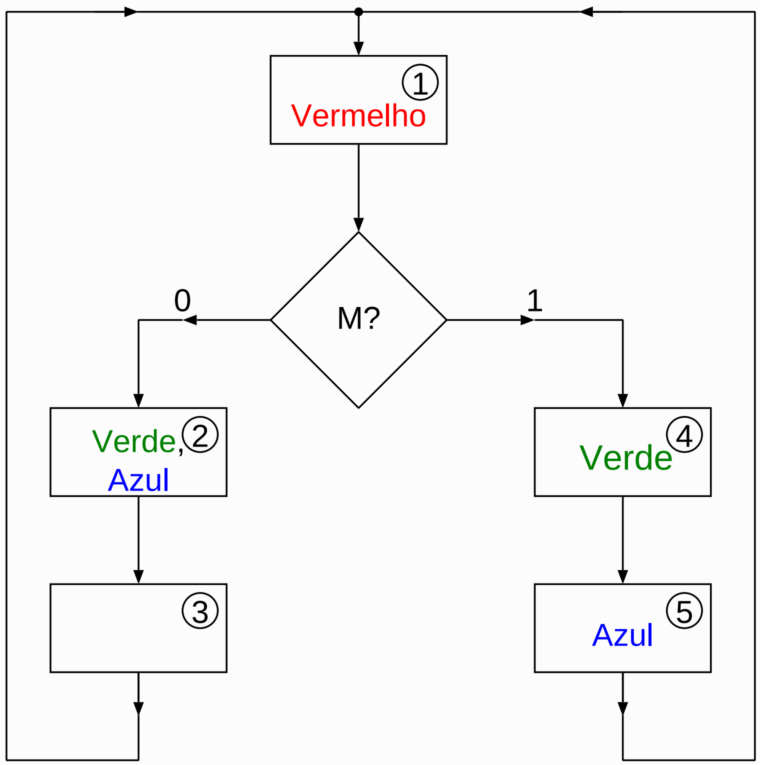
Exemplo de Controladores Digital com um Flip-Flop por estado:
Projecte um circuito que estabeleça a seguinte sequência de controlo num sistema de luzes com uma lâmpada Vermelha, Azul e Verde:
O sistema tem uma variável **M** que controla a sequência. da seguinte forma:

Se **M**=0 ⇒ Vermelho → Verde+Azul → (Tudo apagado) → (Volta ao princípio)

Se **M**=1 ⇒ Vermelho → Verde → Azul → (Volta ao princípio)

Fluxograma

Estabelece a sequência de controlo pretendida

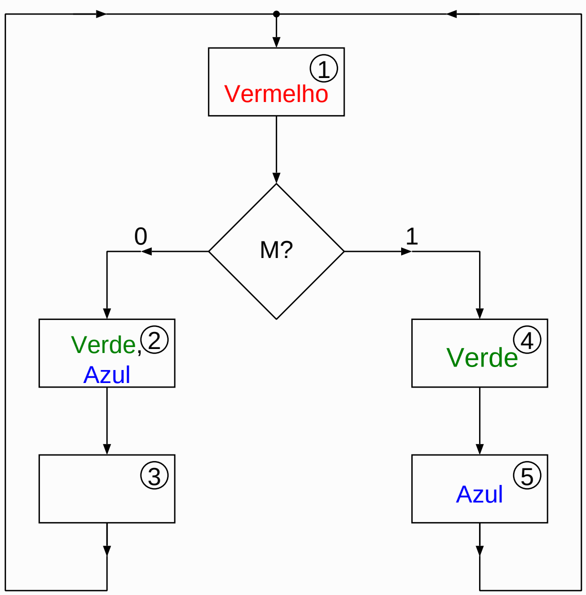


Controladores Digitais

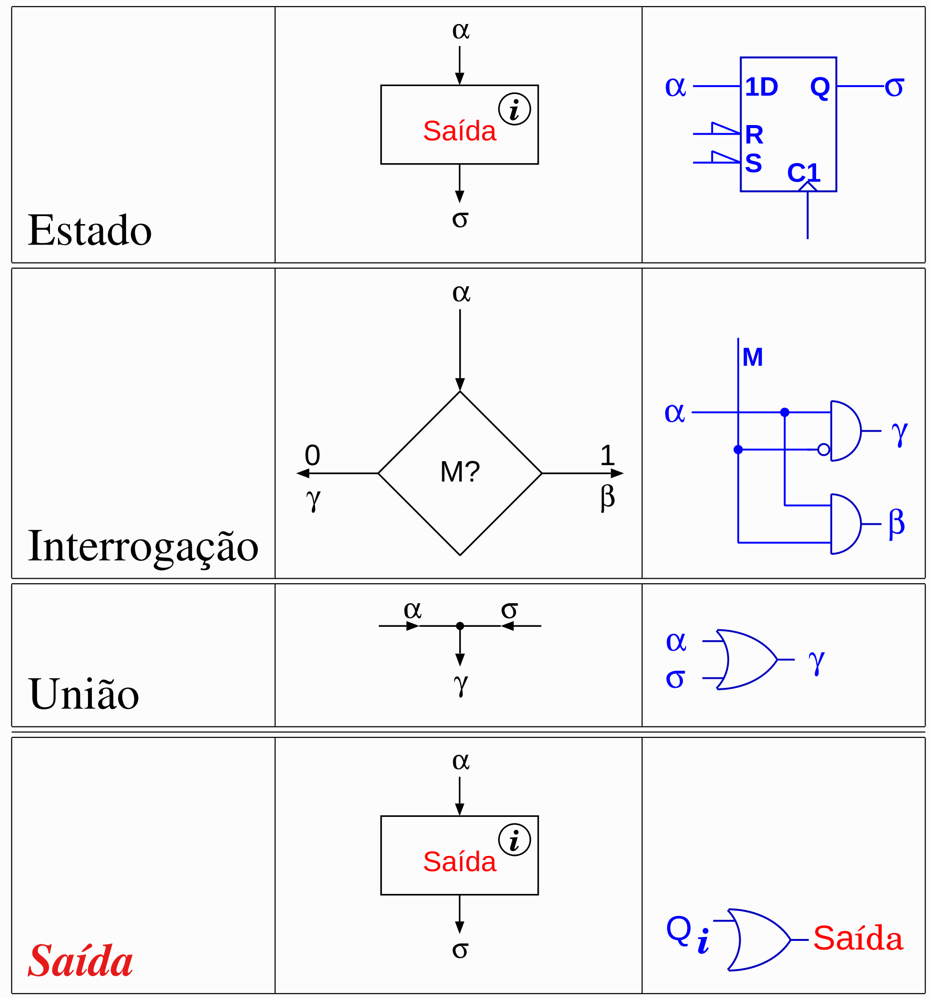
Exemplo de Controladores Digital com um Flip-Flop por estado:
Projecte um circuito que estabeleça a seguinte sequência de controlo num sistema de luzes com uma lâmpada Vermelha, Azul e Verde:
O sistema tem uma variável **M** que controla a sequência da seguinte forma:

Se **M**=0 ⇒ Vermelho → Verde+Azul → (Tudo apagado) → (Volta ao princípio)

Se **M**=1 ⇒ Vermelho → Verde → Azul → (Volta ao princípio)

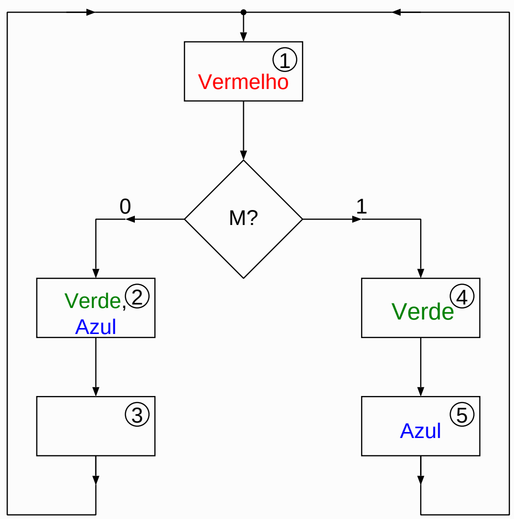


Controlador Digital Sequencial

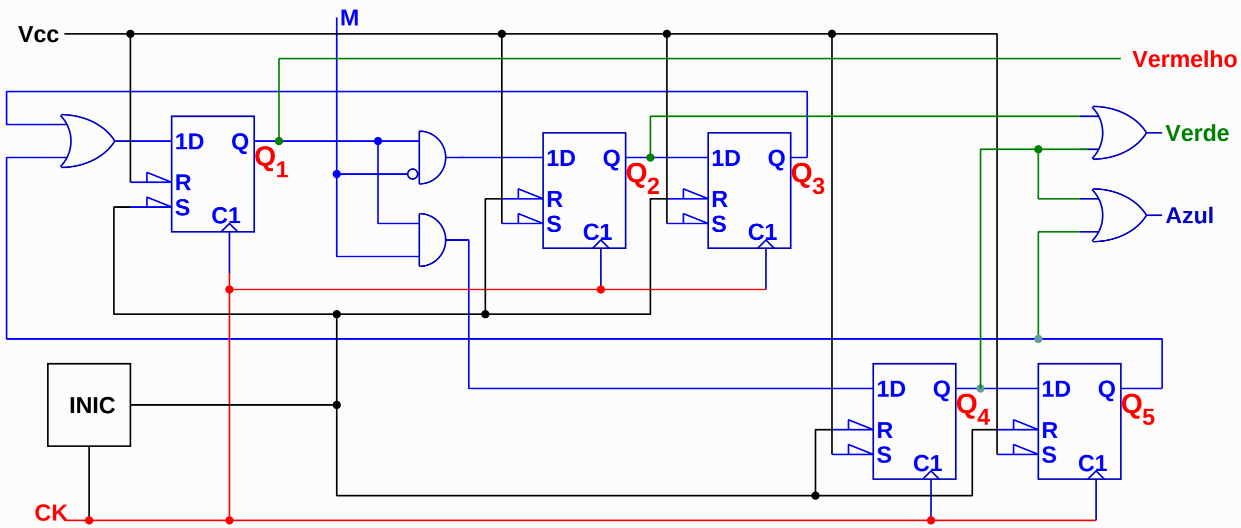


Controladores Digitais

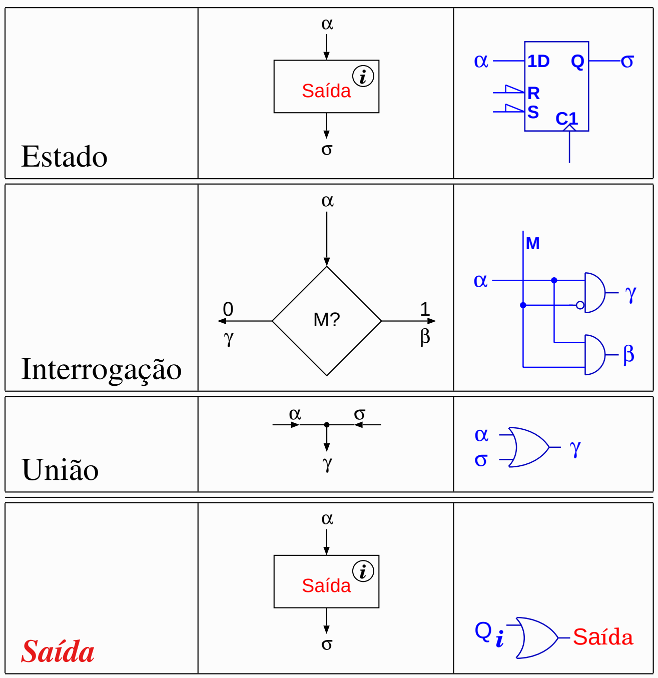
Fluxograma

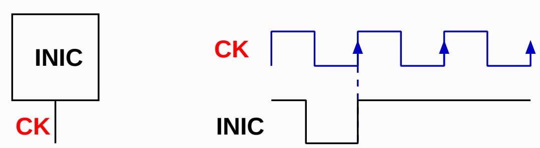


Circuito



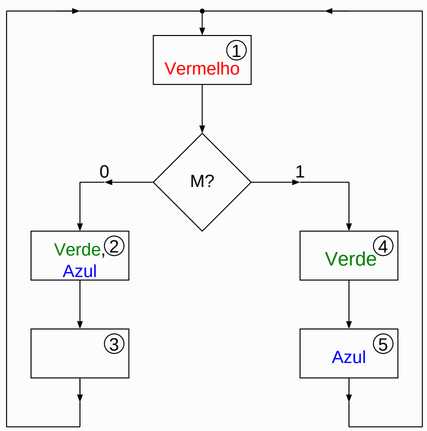
Controlador Digital Sequencial

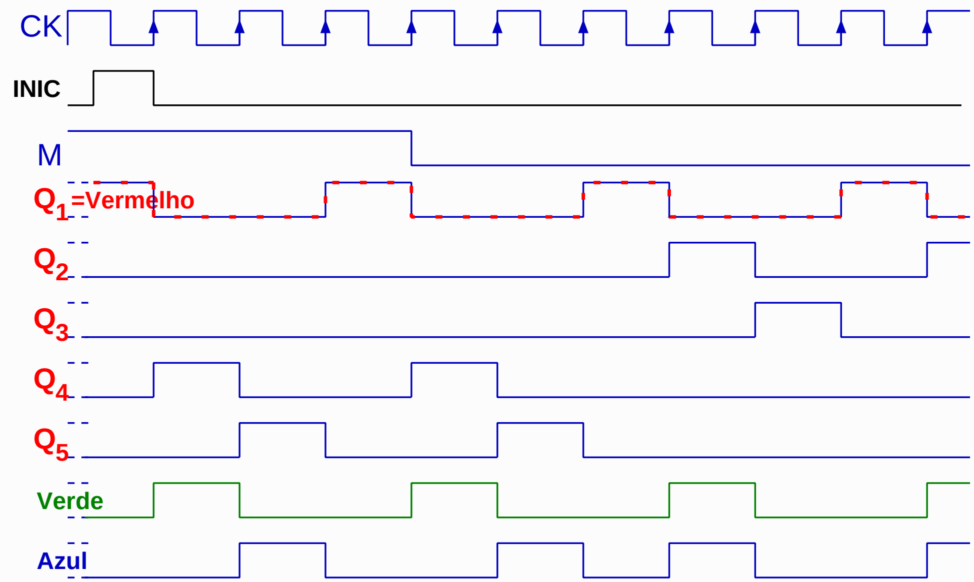




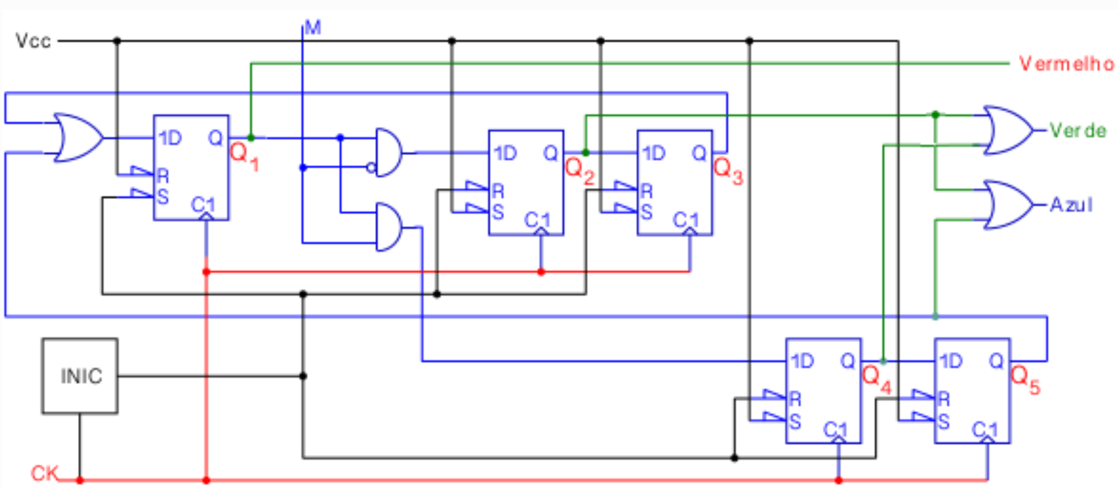
Controladores Digitais

Fluxograma





Circuito



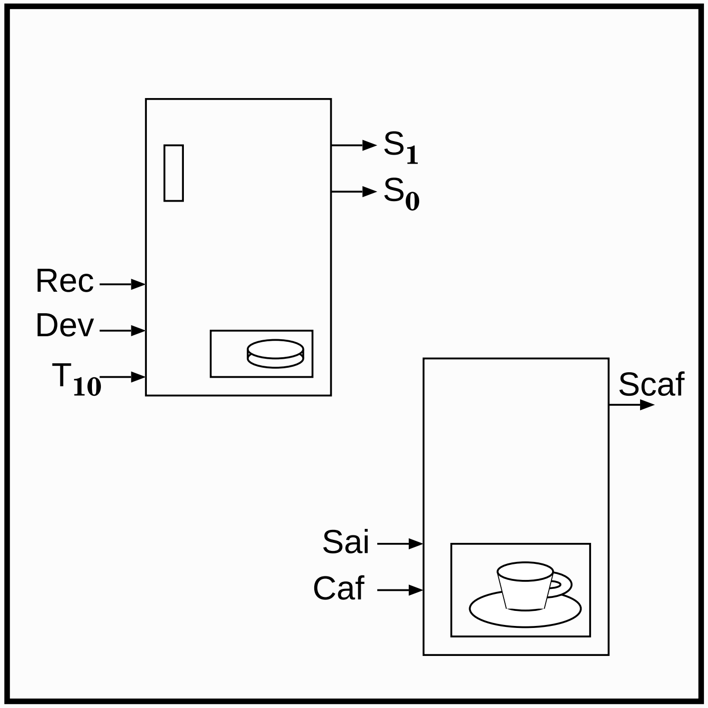
Controladores Digitais

Exemplo de Controlador

Projecte um controlador digital para o aparelho de tirar cafés da figura. Consoante a moeda que entra vão ser activadas as variáveis S_1 e S_0 de acordo com a tabela. Assim, a máquina aceita moedas de 20 cêntimos e 50 cêntimos. Qualquer outra moeda é devolvida, e um depósito de moedas de 10 cêntimos está disponível para permitir dar troco. O custo de cada Café é 20 cêntimos e quando uma moeda de 50 cêntimos é introduzida, a máquina de café deve produzir dois cafés e dar o respectivo troco. Além disso estão disponíveis as seguintes variáveis de controlo:

Rec - Activa a possibilidade de recolha de moeda
Dev - Activa a devolução da moeda introduzida
T_{10} - Activa a devolução de uma moeda de 10 cêntimos
Sai - Activa a saída do Café
Caf - Activa a produção de um café

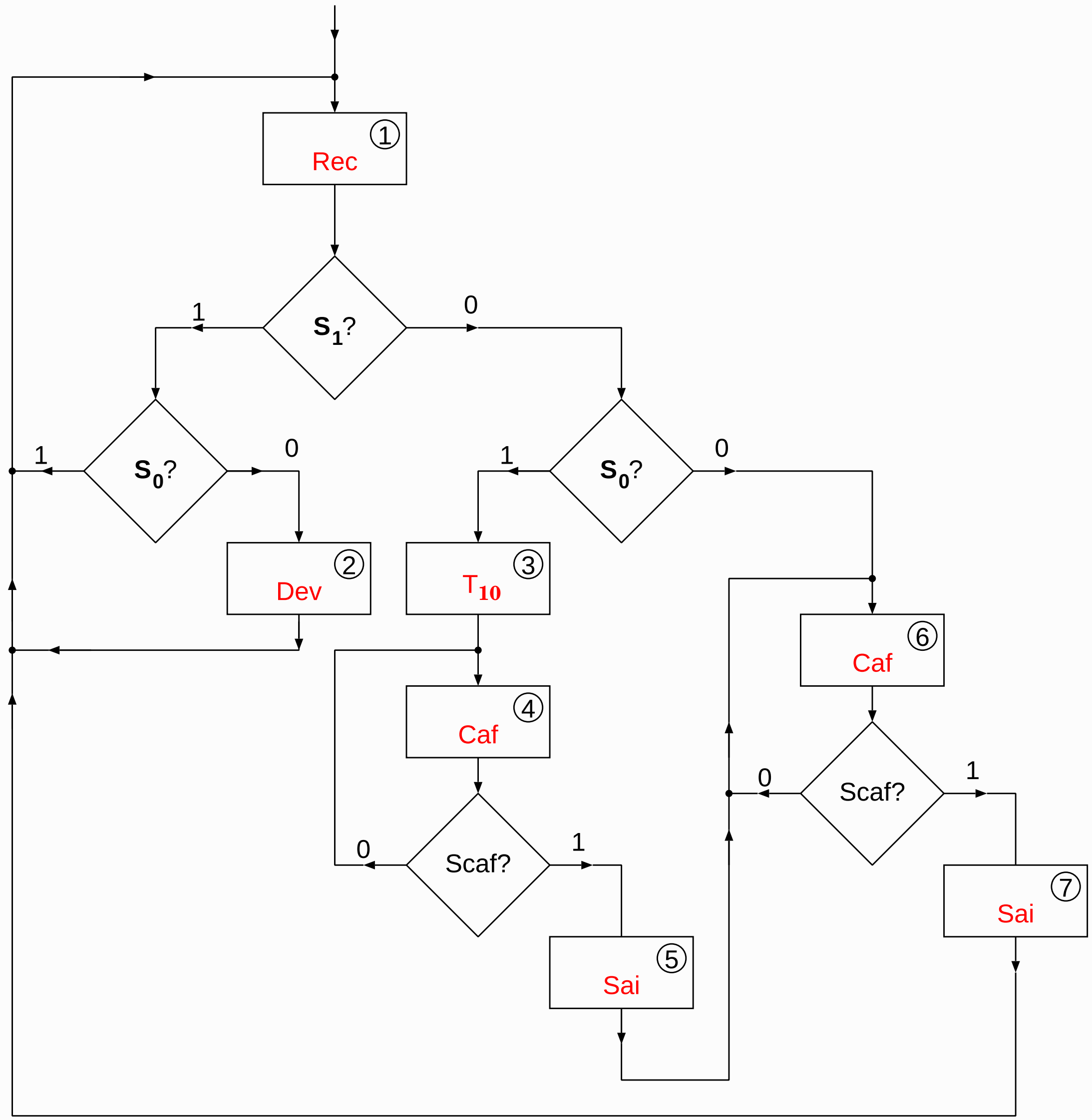
Para controlar o tempo de produção de um café o sistem disponibiliza uma variável lógica Scaf que quando a UM lógico define que o café já está pronto.



S_1	S_0	Significado
0	0	Moeda de 0.20
0	1	Moeda de 0.50
1	0	Moeda $\neq$ 0.20 e 0.50
1	1	Nenhuma Moeda

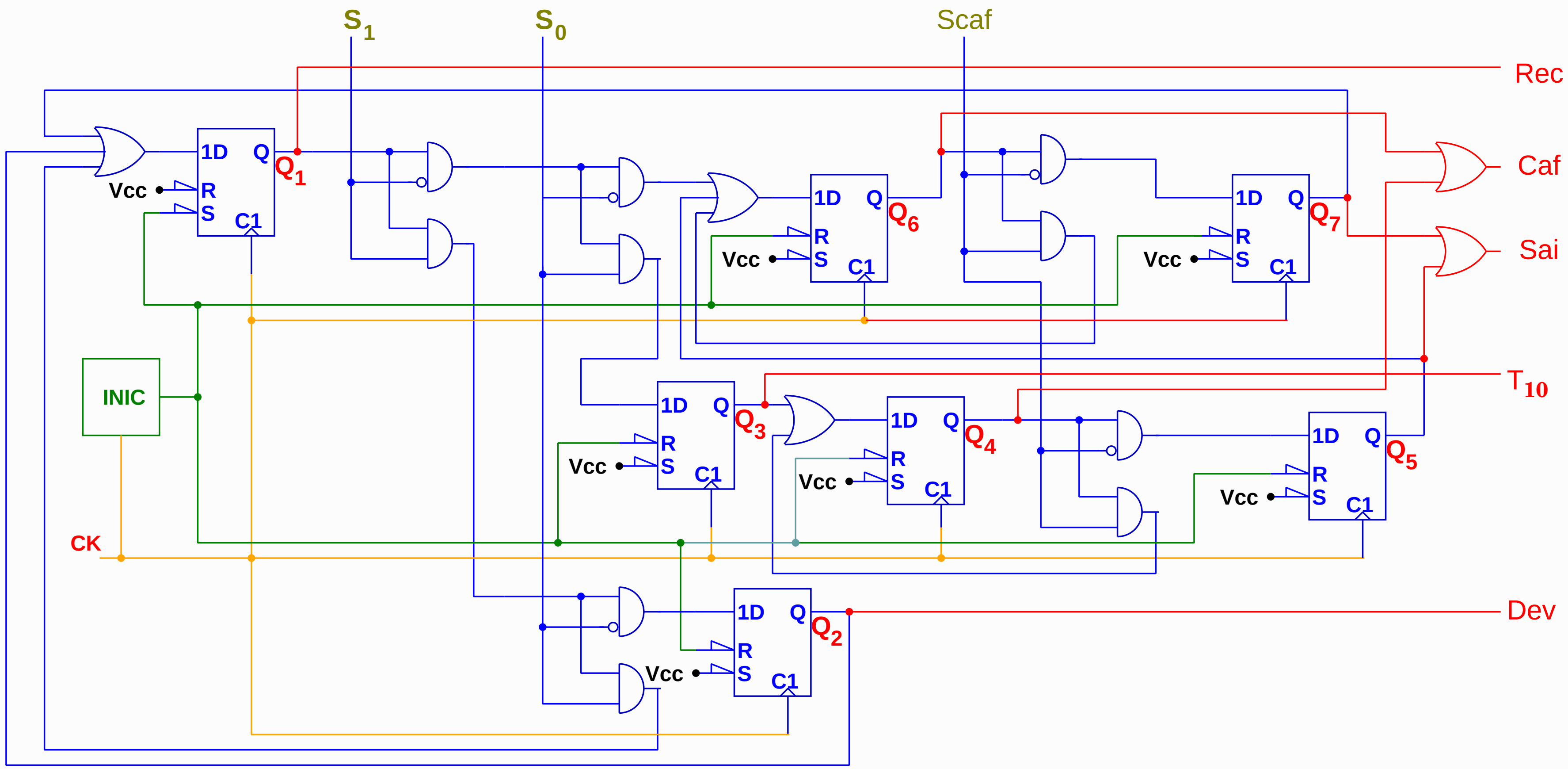
Controladores Digitais

Fluxograma



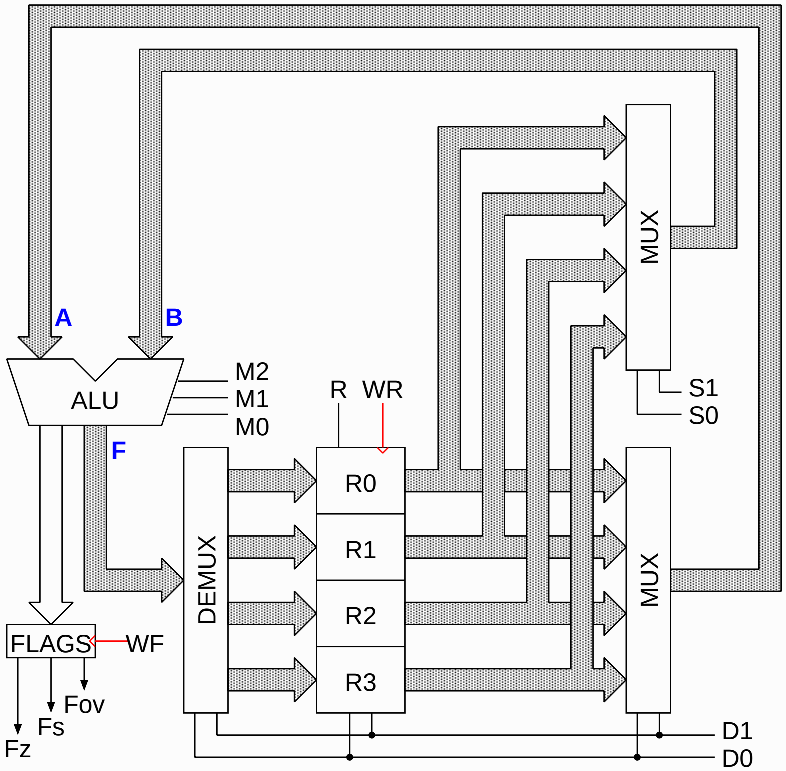
Controladores Digitais

Circuito



Controladores Aritméticos

Arquitectura Aritmética



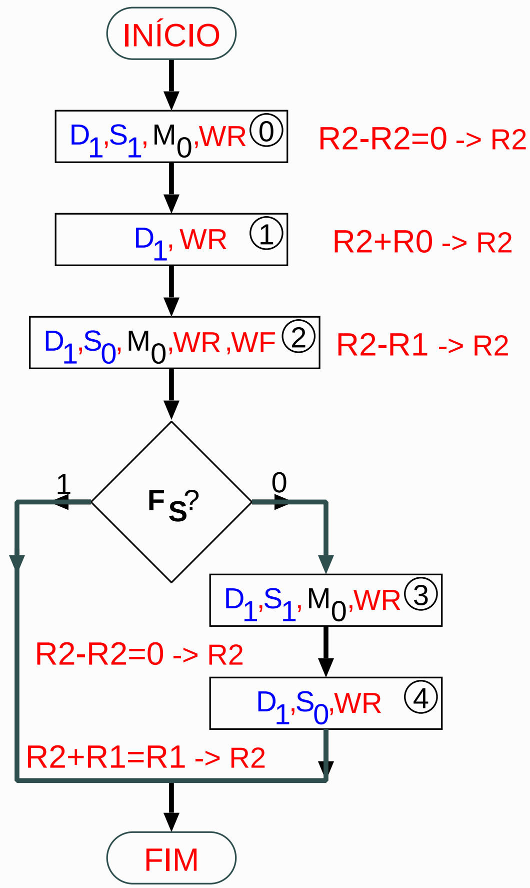
M_2	M_1	M_0	F
0	0	0	A + B
0	0	1	A - B
0	1	0	A + 1
0	1	1	A - 1
1	0	0	A · B
1	0	1	A + B
1	1	0	A ⊕ B
1	1	1	$\overline{A}$

Nota:
M_2=0 - Operações Aritméticas
M_2=1 - Operações Lógicas

F_S - Flag de Sinal
(Bit mais significativo do Acumulador).
F_Z - Flag de Zero (=$\overline{\mathbf{F}_{N-1}+...+\mathbf{F}_1+\mathbf{F}_0}$).
F_{Ov} - Flag de "Overflow"(=$\mathbf{C}_N \oplus \mathbf{C}_{N-1}$).
N - Dimensão da palavra binária da arquitectura.

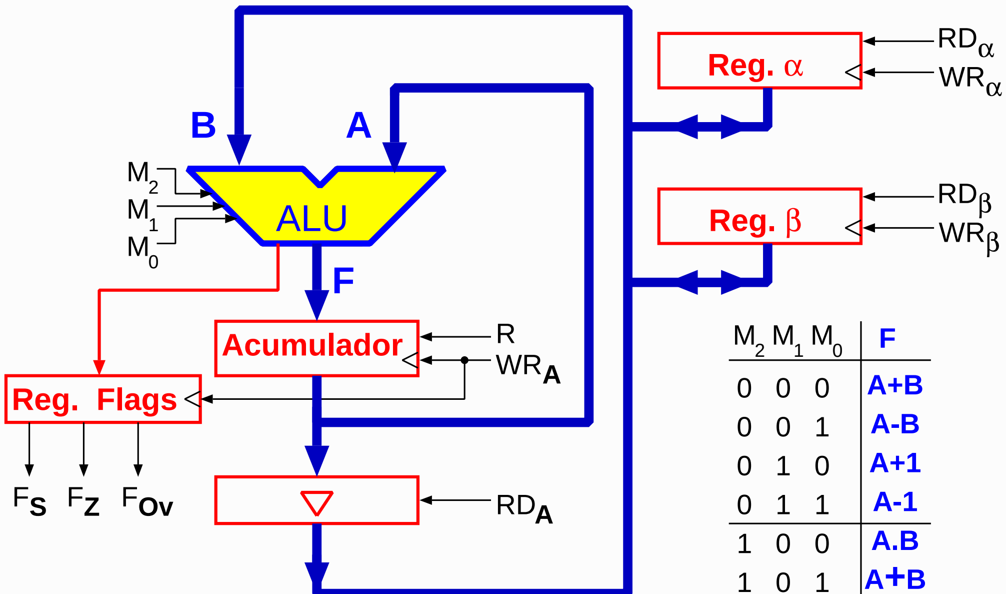
Controlador que faça a seguinte operação:

$$R_2 = \begin{cases} R_1 & se \;\; R_0 \geq R_1 \\ R_0 - R_1 & se \;\; R_0 < R_1 \end{cases}$$



Controladores Aritméticos

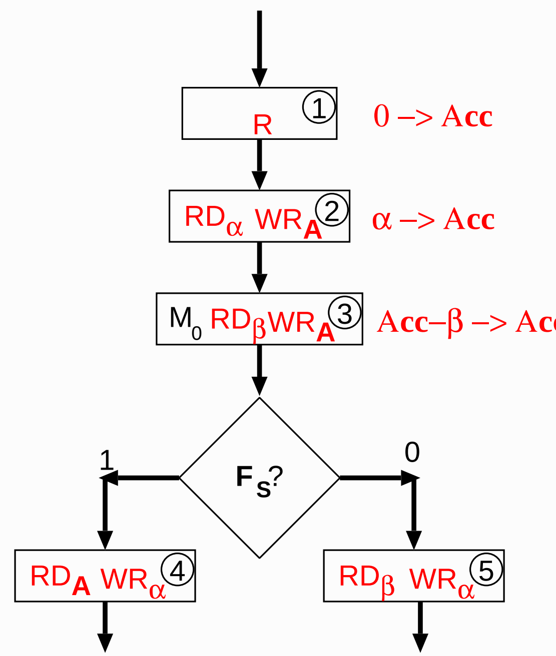
Arquitectura



M_2	M_1	M_0	F
0	0	0	A+B
0	0	1	A-B
0	1	0	A+1
0	1	1	A-1
1	0	0	A.B
1	0	1	A+B
1	1	0	A⊕B
1	1	1	$\overline{A}$

Controlador que faça a seguinte operação:

$$\alpha = \begin{cases} \beta & se \;\; \alpha \geq \beta \\ \alpha - \beta & se \;\; \alpha < \beta \end{cases}$$



F_S - Flag de Sinal
(Bit mais significativo do Acumulador).

F_Z - Flag de Zero ($=\overline{\mathbf{F}_{N-1}+\ldots+\mathbf{F}_1+\mathbf{F}_0}$).

F_{Ov} - Flag de "Overflow"($=\mathbf{C}_N \oplus \mathbf{C}_{N-1}$).

N - Dimensão da palavra binária da arquitectura.