

## 2.º Trabalho Prático

### Introdução:

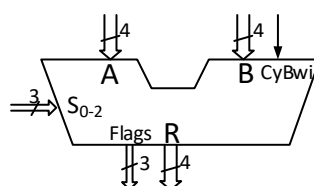
A generalidade das linguagens de programação disponibiliza os seguintes operadores básicos:

- **Aritméticos**                    +, -, \*, /, %;
- **Lógicos**                        &, |, ^, !, ~;
- **Deslocamento (Shift)**      >>, <<;
- **Relacionais**                    =, >, <, >=, <=.

Para suportar a realização destas operações, a Unidade Central de Processamento (CPU) dos nossos computadores inclui na sua arquitectura uma unidade funcional denominada por *Arithmetic Logic Unit* (ALU).

### Objectivo:

Projectar e realizar uma ALU, segundo o diagrama da Figura 1, com as seguintes características:



**Figura 1 – ALU.**

- Executa uma de seis operações (ADD, ADC, SBB, ANL, ASR, RCL), seleccionadas pelos três bits  $S_2$ ,  $S_1$  e  $S_0$ , conforme indicado na Tabela 1 (a codificação poderá ser modificada se for considerada conveniente);

$S_{0-2}$	Operação		Sigla	Z	CyBwo	OV
000	Adição	$R = A + B$	ADD	•	•	•
001	Subtracção com <i>Borrow</i>	$R = A - B - CyBwi$	SBB	•	•	•
010	Adição com <i>Carry</i>	$R = A + B + CyBwi$	ADC	•	•	•
100	Lógico AND bit a bit	$R = A \& B$	ANL	•	–	–
110	<i>Arithmetic Shift Right</i>	$R = A \gg B_{0-1}$	ASR	•	•	–
111	<i>Rotate Carry Left</i>	$R = A \ll 1$	RCL	•	•	•

**Tabela 1 – Funcionalidade da ALU.**

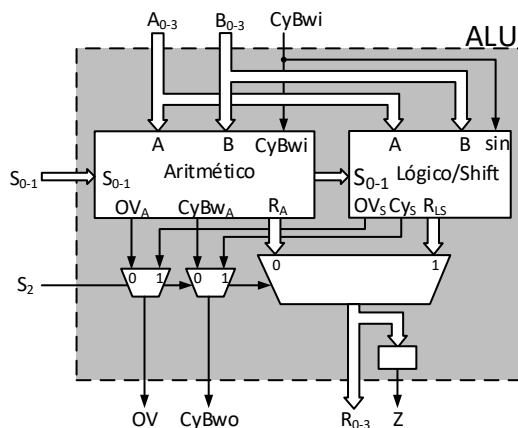
**Legenda:** • : indicador binário com significado na operação, - : indicador binário sem significado na operação.

- Tem por entradas dois operandos A e B, de quatro bits cada, e um operando de um bit, CyBwi (*Carry/Borrow in*), a ser considerado nas operações de adição, subtracção e deslocamento;
- O resultado R é expresso em quatro bits, no mesmo domínio que os operandos;
- A ALU implementa três indicadores binários (*flags*) Z, CyBwo e Ov, sendo um deles qualitativo e os outros de excesso de domínio. Os três indicadores têm o seguinte significado:

Z	<i>Zero</i>	Activo, quando a operação realizada tem como resultado o valor zero.
CyBwo	<i>Carry/Borrow out</i>	Representa <i>Carry</i> na operação de adição e <i>Borrow</i> na operação de subtracção, estando activo quando o resultado excede o domínio, entendida a operação em código natural. Na operação ASR recebe o último bit deslocado e na operação RCL recebe o bit de maior peso de A.
Ov	<i>Overflow</i>	Activo, quando o resultado da adição ou da subtracção excede o domínio, entendida a operação em código dos complementos. Nas operação RCL quando o bit de sinal de R difere do bit de sinal de A.

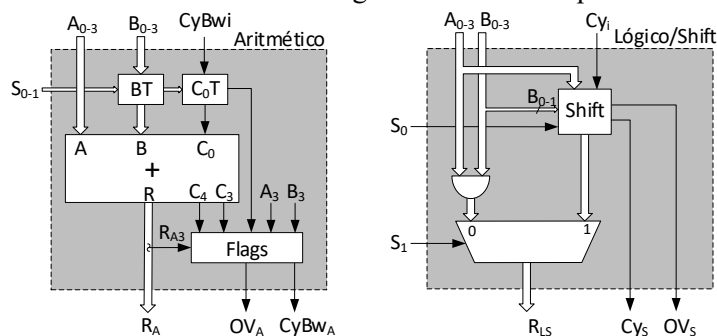
### Realização:

A ALU a desenvolver deve ser constituída por dois módulos, interligados conforme o diagrama de blocos apresentado na Figura 2. O módulo Aritmético realiza as operações ADD, ADC e SBB, enquanto o módulo Lógico/Shift realiza as operações ANL, ASR e RCL.



**Figura 2 – Diagrama de blocos da ALU.**

A arquitectura interna dos dois módulos obedece aos diagramas de blocos apresentados na Figura 3.



**Figura 3 – Diagramas de blocos dos módulos Aritmético e Lógico/Shift.**

### Módulo Aritmético:

- O módulo aritmético poderia ser realizado por multiplexagem dos operadores adição e subtração. A arquitectura preconizada implementa uma técnica denominada por *contraction* (contração) que consiste em reutilizar um dado elemento funcional para a realização de várias funções, obtendo-se uma estrutura mais simples, por adaptação das respectivas entradas e saídas;
- Neste módulo, o elemento central é o somador completo de quatro bits;
- Os elementos BT e C0T realizam a transformação dos operandos B e CyBwi para que, utilizando o elemento somador, se realizem as operações de adição e subtração;
- O elemento Flags, implementa os sinais binários OV<sub>A</sub> e CyBw<sub>A</sub>. A sua implementação poderá recorrer a outros sinais disponíveis na estrutura, caso o aluno os considere preferíveis.

### Módulo Lógico:

- A arquitectura preconizada recorre à multiplexagem do resultado dos vários operadores, sendo estes realizados por módulos independentes entre si.
- As operações de deslocamento (ASR e RCL) são realizadas por um *Barrel Shifter*. Na operação ASR o número de bits a deslocar é determinado pelos bits B<sub>0-1</sub>. Nesta operação, o bit Cy<sub>S</sub> recebe o último bit deslocado. Na operação RCL, o bit Cy<sub>S</sub> recebe o bit A<sub>3</sub> e A<sub>0</sub> recebe o bit CyBwi.

## Lógica e Sistemas Digitais

Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores

O trabalho é realizado em grupo e a apresentação decorre em laboratório, em data a combinar com o docente.

Antes de passar à realização da montagem, sobre *breadboard*, devem ser elaborados os seguintes elementos:

- o diagrama de blocos correspondente à estrutura da solução adoptada;
- implementação em logisim;
- a programação da PAL, em linguagem CUPL;

Antes da apresentação, as montagens devem ser testadas nas bases de montagem existentes no laboratório.

Durante a apresentação são validados os elementos anteriormente referidos.

Após a apresentação, cada grupo deve elaborar um relatório sobre o trabalho realizado, do qual conste, para além dos documentos acima indicados:

- o enunciado do trabalho;
- a descrição sucinta dos métodos adoptados no projecto;
- Esquema logisim e listagem CUPL.