

Projeto Final: Unidade Lógica Programável

1 Introdução

O projeto é de uma Unidade Lógico-Aritmética (ULA) simples. Uma ULA é um circuito responsável por realizar operações aritméticas e lógicas em um sistema digital, onde a operação que deve ser executada é determinada por sinais de controle externos. Uma representação simplificada de uma ULA está apresentada na Figura 1↓, onde **A** e **B** são os dados (ou operandos) sobre os quais será realizada uma operação. Como há várias opções para as operações, há uma seleção de qual deve ser a operação, indicada pela entrada **S**. O resultado da operação é apresentado na saída da ULA.

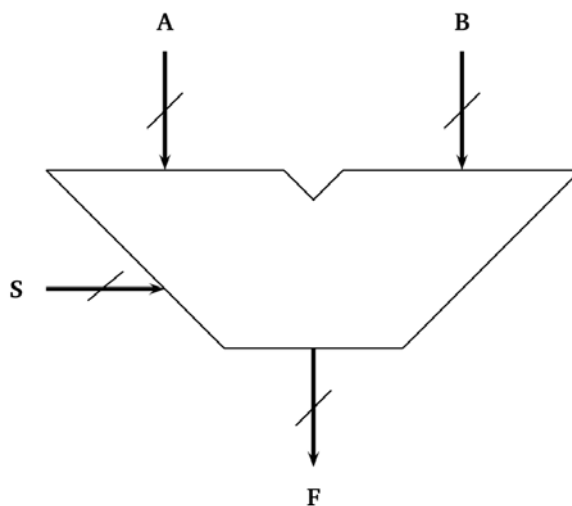


Figure 1 Esquema simplificado de uma ULA.

2 Projeto Básico

O sistema deve possuir as seguintes características:

Comprimento dos operandos A e B: 4 bits ($A_0 A_1 A_2 A_3$) e ($B_0 B_1 B_2 B_3$).

Tipos de operação: há dois tipos de operação - com um ou com dois operandos, sendo as operações lógicas ou aritméticas. As operações são selecionáveis por um comando **S** de 4 bits e o resultado **F** é calculado de acordo com Tabela 1↓.

Tabela 1. Operações a serem executadas

Mnem.	$S_3 S_2 S_1 S_0$	$F_3 F_2 F_1 F_0$	Mnem.	$S_3 S_2 S_1 S_0$	$F_3 F_2 F_1 F_0$
ZERA	0 0 0 0	0 0 0 0	SOMA(A,B)	1 0 0 0	A+B
TUDOUM	0 0 0 1	1 1 1 1	SUBT(A,B)	1 0 0 1	A-B
OPA	0 0 1 0	$A_0 A_1 A_2 A_3$	MULT(A,B)	1 0 1 0	A.B
OPB	0 0 1 1	$B_0 B_1 B_2 B_3$	DIV(A,B)	1 0 1 1	A/B
OR(A,B)	0 1 0 0	OR bit a bit	MOD(A,B)	1 1 0 0	A mod B
AND(A,B)	0 1 0 1	AND bit a bit	SQA(A)	1 1 0 1	A^2
XOR(A,B)	0 1 1 0	XOR bit a bit	NEGA	1 1 1 0	-A
NOTA	0 1 1 1	NOT bit a bit	ADD1	1 1 1 1	A+1

Se operações resultarem em *overflow* aritmético, o *display* deve apresentar a palavra “Erro”.

Saída de Dados: o resultado das operações deve ser apresentado nos quatro *displays* de sete segmentos. Há quatro modos de apresentação, selecionáveis por duas chaves $C_1 C_0$: decimal (00), hexadecimal (01), binário (10) e octal (11).

Modos de Operação: há dois modos de uso do sistema, selecionados por uma chave. No modo “Calculadora” (CALC), os operandos A e B, assim como o seletor de operações S, são inseridos por chaves.

No modo “Programa” (PROG), uma sequência de operações é armazenada em uma memória e o resultado da operação corrente é armazenado em um registrador chamado de “Acumulador” (ACC). O valor do acumulador é visualizado no *display*.

O procedimento para inserção das operações na memória é simples: cada operação é definida por chaves e inserida em posições sequenciais de memória, por intermédio de um botão “Enter”.

Cada vez que o botão “Enter” é pressionado, um contador chamado “Endereço de Memória” (EDM) incrementa de uma unidade o valor de sua saída, de modo a indicar a posição de memória onde a operação será armazenada. A memória tem 1024 posições (10 bits), e cada uma armazena uma operação de 4 bits. A posição inicial de memória é sempre

zero. A cada passo, a posição de memória corrente (seu endereço) é apresentada em hexadecimal nos três primeiros *displays*, e o código da operação a ser inserida no quarto *display* (também em hexadecimal).

Após inserida a última operação, o botão “Run” inicia a execução do programa.

2.1 PseudoAlgoritmo do programa

1. Escolha modoULA = CALC ou modoULA = PROG
2. Caso modoULA = CALC
 1. Escolha:
 1. operação (4 chaves)
 2. base numérica de apresentação do resultado (4 chaves)
 3. as entradas A e B (4 chaves)
 2. Se resultado for *overflow* aritmético
 1. Display mostra “ERRO”
 2. caso contrário
 1. display mostra o resultado
3. Caso modoULA = PROG (a qualquer momento, o botão LIMPA limpa o ACC)
 1. Se botão RUN é apertado, então
 1. As operações armazenadas em ACC são executadas da seguinte maneira:
 1. a primeira operação em ACC é executada com as entradas sendo as chaves A e B e o resultado é mostrado no display (se tiver overflow, coloque ERRO e limpe o acumulador)
 2. Se o botão CONTINUA for apertado, então a próxima operação é executada. Ela é feita com o resultado anterior sendo a chave A (no lugar dele) e a chave B.
 3. Repita o procedimento B até que:

1. As operações acabarem. Neste caso, ACC é limpo e o programa volta ao início de CALC
2. ou modoULA = CALC

2. caso contrário:

1. Escolha uma operação (4 chaves)

1. Aperte o botão Armazenar
2. o número da operação é armazenado no acumulador
3. os 3 primeiros displays mostram o endereço da memória em que essa operação foi armazenada em hexadecimal
4. o quarto display mostra o resultado em decimal

3 Grupos

Os projetos devem ser feitos em grupos de 3 alunos.

4 Avaliação

O grupo deve enviar um arquivo compactado (p.ex., nos formatos .zip, .rar) na Tarefa correspondente disponibilizada na sala do Moodle desta disciplina. Nesse arquivo compactado, devem estar:

1. a pasta do projeto criada no Vivado com os arquivos da ULA (não esqueça de incluir o arquivo de restrições), e
2. um arquivo texto com os nomes e matrículas dos integrantes do grupo.

Além disso, os alunos devem apresentar a Basys3 funcionando com o código criado no dia **06/07/2018** em sala de aula. Além disso, o código deve ser enviado até às 11:55 desse mesmo dia, pelo Moodle.