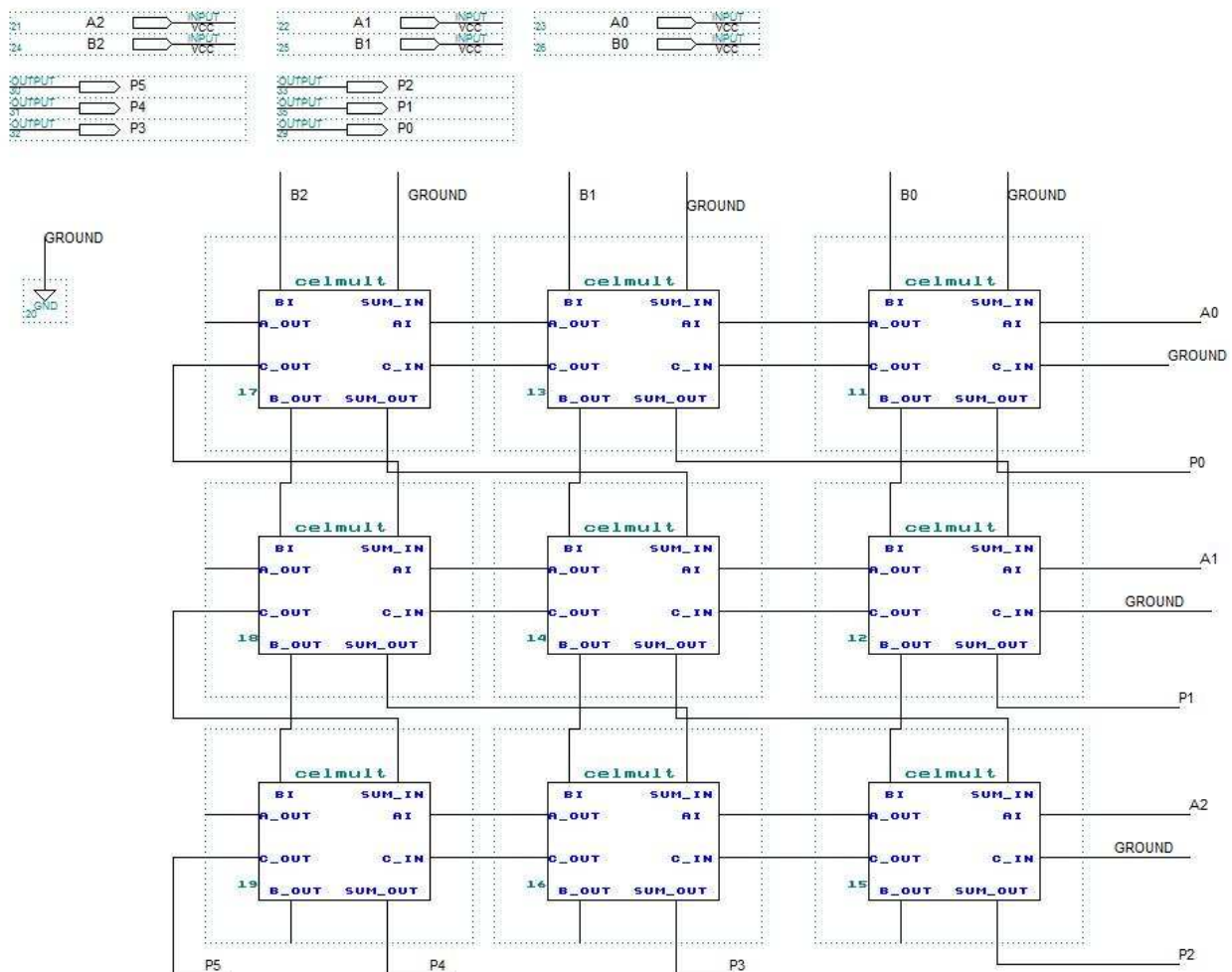


Introdução

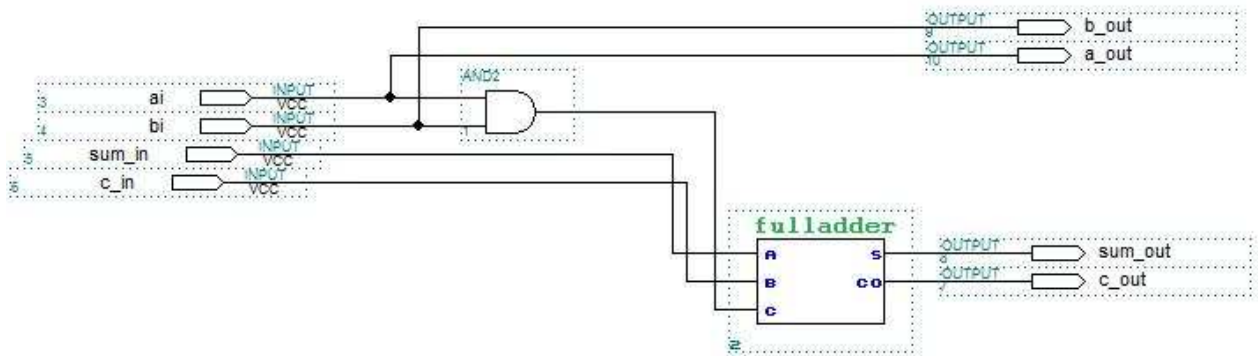
O trabalho prático do dia 14/05 teve como objetivo projetar, implementar e simular multiplicadores paralelos de 3*3 bits, usando soma e deslocamento e estrutura tipo matriz.

As atividades dividiram-se em 3 partes: a primeira consistia em projetar, implementar e simular um multiplicador paralelo usando soma e deslocamento, tendo entradas A e B de 3 bits e saída S de 6 bits. A segunda parte difere na segunda apenas pelo multiplicador paralelo ser do tipo matriz. A terceira parte objetivava simulações funcionais e temporais em ambos projetos e conclusões a respeito da complexidade, tempo de resposta e modularidade das duas técnicas de implementação.

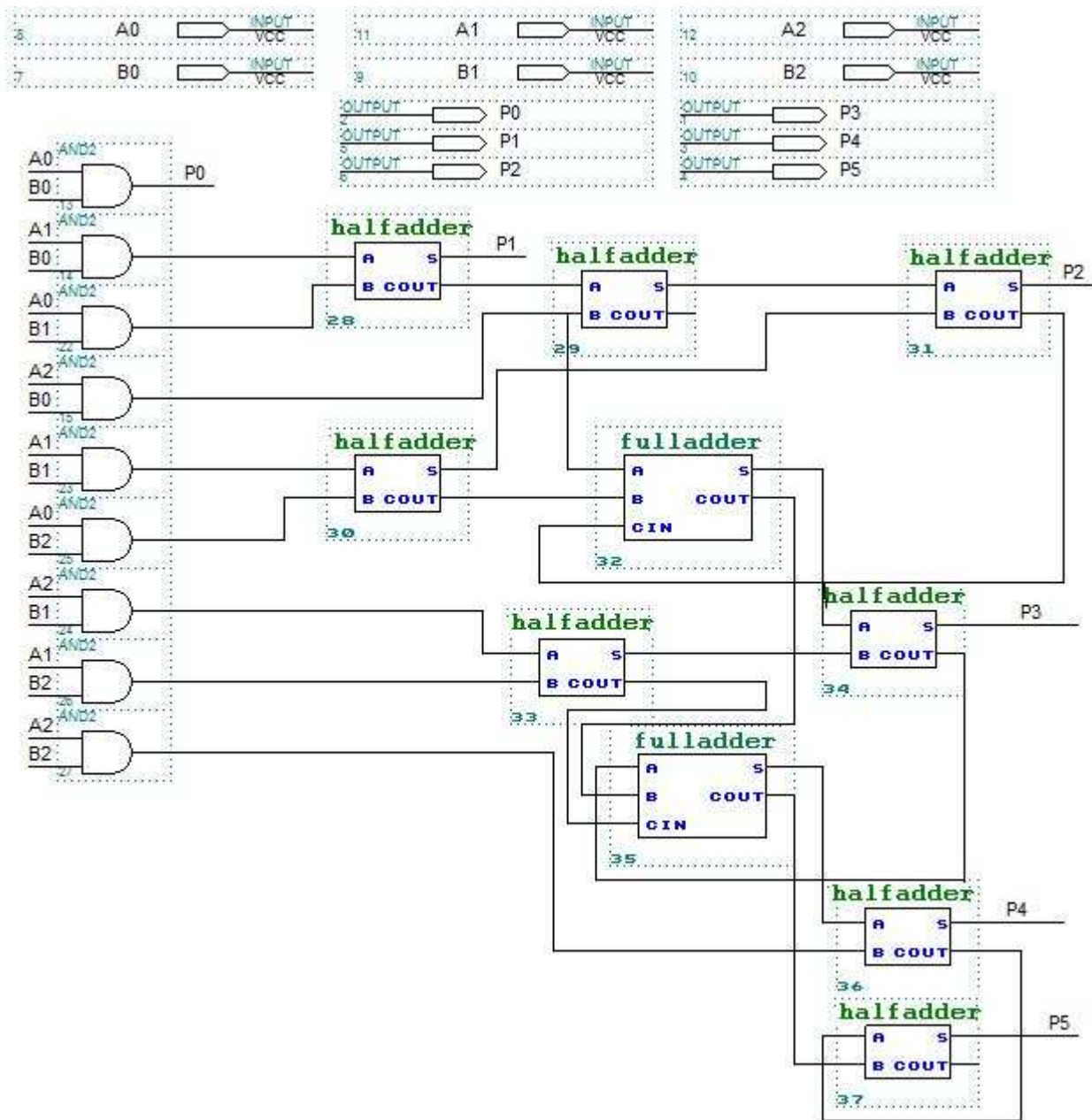
Diagrama em Blocos



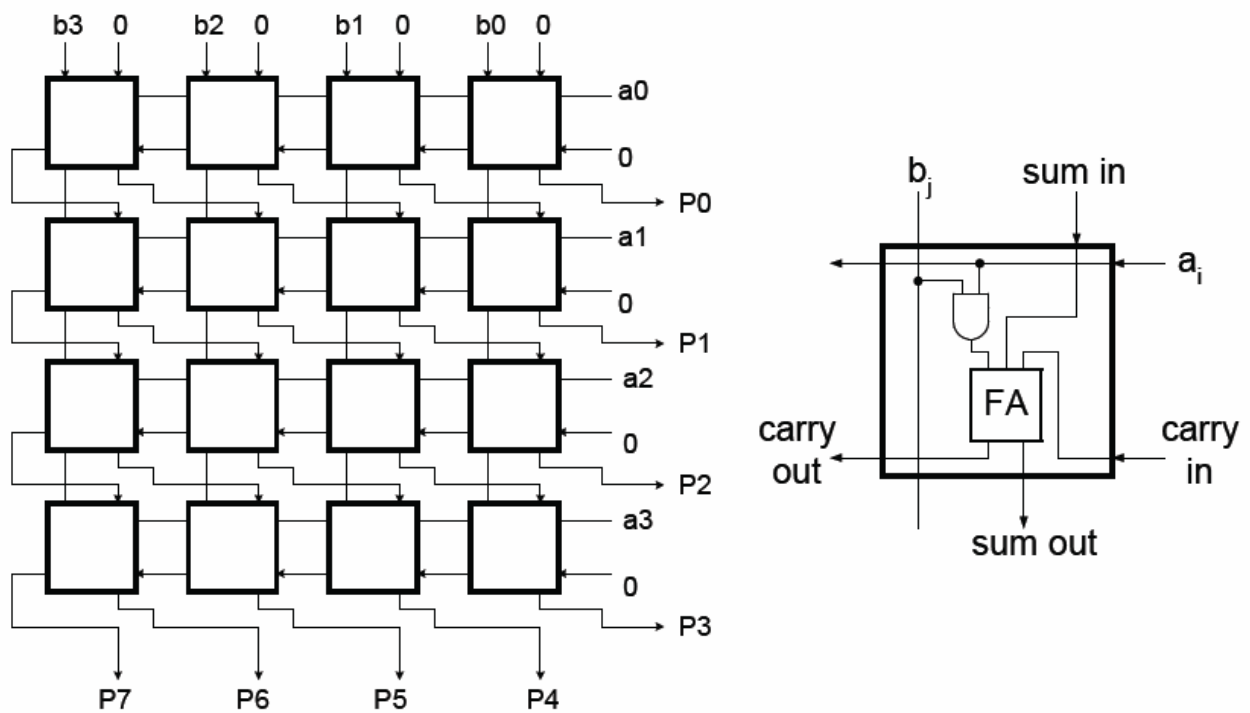
Multiplicador 3x3 usando estrutura do tipo matriz



Célula do multiplicador em matriz



Multiplicador 3x3 usando soma e deslocamento



Algoritmo multiplicador estrutura tipo matriz

			A2	A1	A0
		x	B2	B1	B0
Cout(P4)	A2B2 Cout(P3)	A2B1 A1B2 Cout(P2)	A2B0 A1B1 A0B2 Cout(P1)	A1B0 A0B1	A0B0
P5	P4	P3	P2	P1	P0

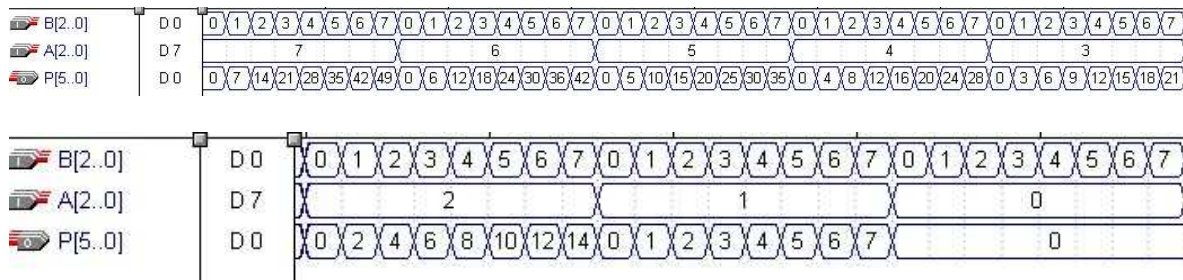
Algoritmo multiplicador soma e deslocamento

Simulações

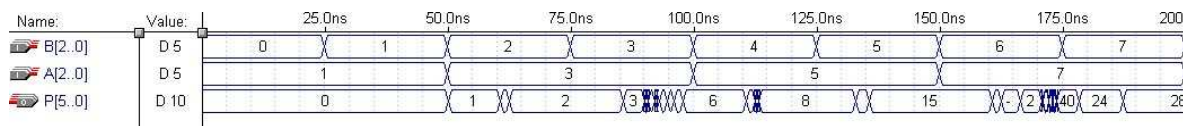
A seguir, são apresentadas as simulações funcionais e temporais para os dois multiplicadores:

Multiplicador Paralelo 3*3 bits usando soma e deslocamento:

Simulação Funcional:

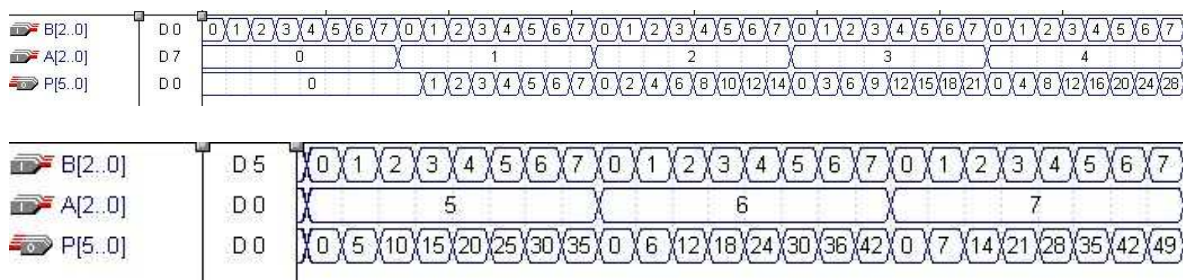


Simulação Temporal:

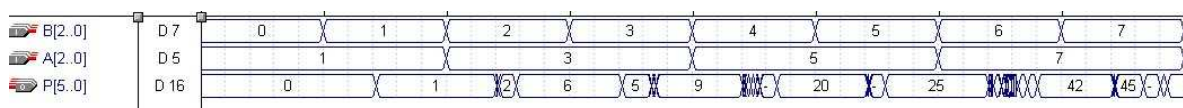


Multiplicador Paralelo 3*3 bits usando estrutura tipo matriz:

Simulação Funcional:



Simulação Temporal:



Análise das duas técnicas

Pode-se observar que o multiplicador em matriz apresenta uma considerável complexidade, se comparado ao multiplicador por somas e deslocamentos, uma vez que as saídas do multiplicador em matriz são diretamente dependentes de todas as entradas (enquanto no multiplicador com somas e deslocamentos são utilizadas funções somas 'locais' que passam o *carry* adiante) e, assim, possuem mais portas lógicas em sua composição. Essa maior complexidade é, entretanto, compensada pelos tempos de resposta: como o multiplicador em matriz não é tão dependente de *carry* (apesar de cada nível depender do *sum_in* do nível anterior), o tempo de atraso é amenizado. Percebemos que o multiplicador em matriz é também mais modular do que o combinacional, o que facilitou consideravelmente sua implementação no MaxPlus. A tabela a seguir ilustra a explicação da complexidade:

Circuito	Quantidade de Portas Lógicas		
	AND	OR	XOR
Célula do Multiplicador em Matriz	3	1	2
Multiplicador em Matriz	27	9	18
Multiplicador Combinacional	21	2	12

Conclusão, interesses, dificuldades e sugestões

O trabalho proposto nessa aula prática foi bastante tranquilo. Havia certo tempo que não terminávamos o trabalho inteiramente durante a aula no laboratório e dessa vez conseguimos. E isso foi possível pois o trabalho não era muito extenso mas ainda assim era interessante.

O projeto, implementação e simulação dos dois multiplicadores (soma e deslocamento e estrutura tipo matriz) utilizaram conhecimentos de aulas passadas (half-adder e full-adder), o que contribui para a fixação do conteúdo. Além disso, esse tipo de trabalho é interessante pois, através da simulação, é possível ver saídas que demonstram que tudo que aprendemos na teoria aplicam-se à prática.