



---

---

**2º PROJETO:**  
**SCHEDULING AND RESOURCE SHARING**

---

---

PROJETO DE SISTEMAS DIGITAIS  
DOCENTE: PAULO FLORES

**AFONSO RODRIGUES Nº 67528**  
**AFONSO PEREIRA Nº 78949**  
**MIGUEL RODRIGUES Nº 76176**



# 1 Introdução

Neste segundo projeto de laboratório são explorados os conceitos de gestão de recursos e calendarização, bem como o uso de memórias para receber dados e onde escrever dados.

Com o conhecimento adquirido no laboratório anterior em relação ao funcionamento do software Vivado e à linguagem de descrição de hardware VHDL, é criado um programa que resolva uma iteração de um algoritmo (Método do Gradiente) para encontrar um mínimo local de uma equação polinomial de segundo grau. Os valores de cada variável (A, B, C, Xi e Step) estão presentes em memórias em posições consecutivas de memória, com 10 bits, as quais são acedidas sequencialmente de modo a realizar a primeira iteração de quatro conjuntos diferentes de dados. Cada iteração do algoritmo dá origem a três resultados a guardar (dy, Xn e y), originando um total de 12 valores de 40 bits a guardar. Estes são armazenados em posições sequenciais numa única memória de saída.

Considerando restrições de hardware, este projeto leva a que seja feita uma gestão e optimização dos recursos a utilizar que envolve calendarizar as operações a serem feitas, dando uso a diferentes recursos, como o *ASAP Scheduling* ou a lista de prioridades.

## 2 Scheduling

Considerando apenas uma iteração do algoritmo completo, e admitindo que cada operação tem a duração de um ciclo de relógio, são obtidos os seguintes *data flows* e lista de prioridades:

### 2.1 ASAP Scheduling

*As Soon As Possible Scheduling*: Assim que seja possível realizar uma operação, esta é realizada, mesmo que o seu resultado só seja necessário muito tempo depois.

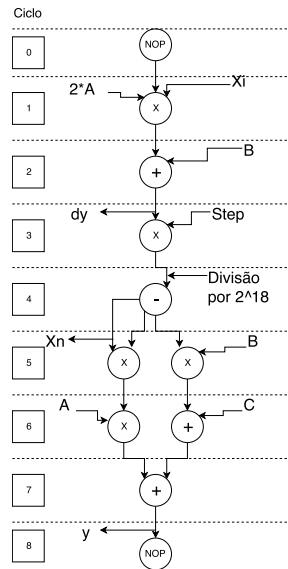


Figura 1: ASAP Scheduling

**Nota:** Tanto a multiplicação por 2 como a divisão por  $2^{18}$  não requerem o uso de recursos adicionais, visto que são ambos múltiplos de 2, constituem apenas *shifts* à esquerda (multiplicação) ou à direita (divisão).

## 2.2 ALAP Scheduling

*As Late As Possible Scheduling*: As operações são realizadas o mais tarde possível sem afetar o número total de ciclos de relógio (em comparação com o caso anterior), isto é, cada operação só se realiza no ciclo anterior a ser necessário o seu resultado.

Neste projeto, este diagrama não vai ser diferente do presente na secção anterior, pois nenhuma operação pode ser realizada mais tarde sem atrasar toda a iteração (e nenhuma poderia ter sido feita mais cedo na secção anterior).

## 2.3 List Scheduling

A lista de prioridades usando o caminho crítico como métrica é a apresentada na Tabela 1. Esta lista tem em conta que os recursos utilizados são os especificados: uma *Arithmetic and Logic Unit* (com a sigla ALU na lista) e dois multiplicadores (com o símbolo X na lista).

Tabela 1: List Scheduling

Nr	Operação	Prioridade
1	X	7
2	ALU	6
3	X	5
4	ALU	4
5	X	3
6	X	3
7	X	2
8	ALU	2
9	ALU	1

## 3 Circuito VHDL

### 3.1 Unidade de Controlo

FSM, recursos, diagrama de estados talvez

### 3.2 Datapath

descrição datapath, o que faz, como faz, que recursos usa etc

### 3.3 Diagrama de Blocos, Recursos e Especificações

Recursos, area clock minimo -> 15 ns freq max... etc

### 3.4 Simulação

## 4 Pipelining

## 5 Conclusão