

DSPs e FPGAs

Prof. Leandro I. Pinto

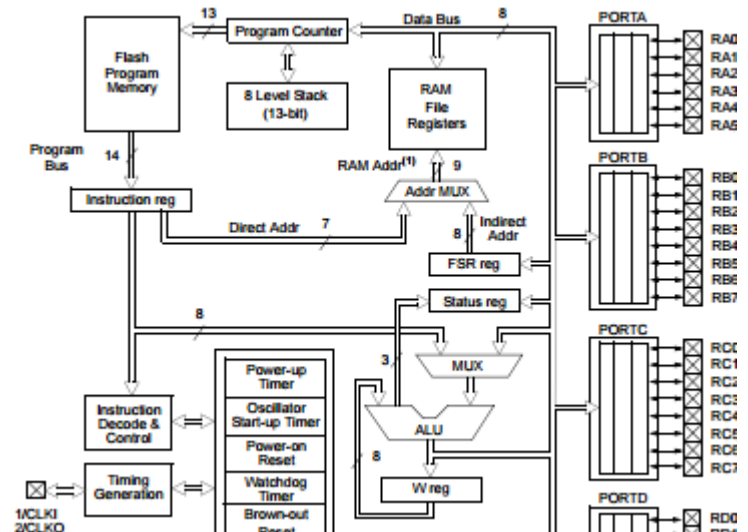
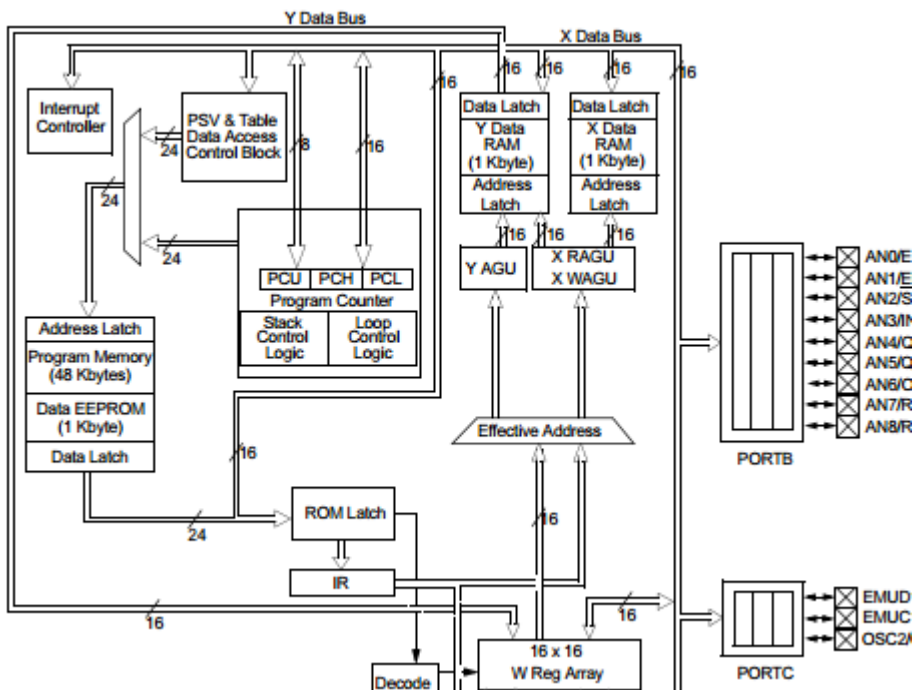
DSPs

- Microprocessador/Microcontrolador especializado;
 - Arquitetura otimizada para operações de processamento de sinais;
- Geralmente utilizado para medir, filtrar e processar sinais contínuos do mundo real;
- Micro de uso geral também podem processar isso
 - Mas DSPs dedicados oferecem maior eficiência em consumo de energia e processamento;
 - Usados em celulares, devido ao menor consumo;



DPSs

- Possuem uma arquitetura diferenciada;
 - Capaz de processar múltiplos dados ou instruções ao mesmo tempo;



DSPs

- Um sistema típico de processamento de sinais;
- Requer alto número de operações matemáticas;
 - Deve ser executado rápido e repetitivamente;
- Atrasos não são toleráveis;
- Muitos Micros. Podem executar esses sinais, mas consomem energia;



DSPs

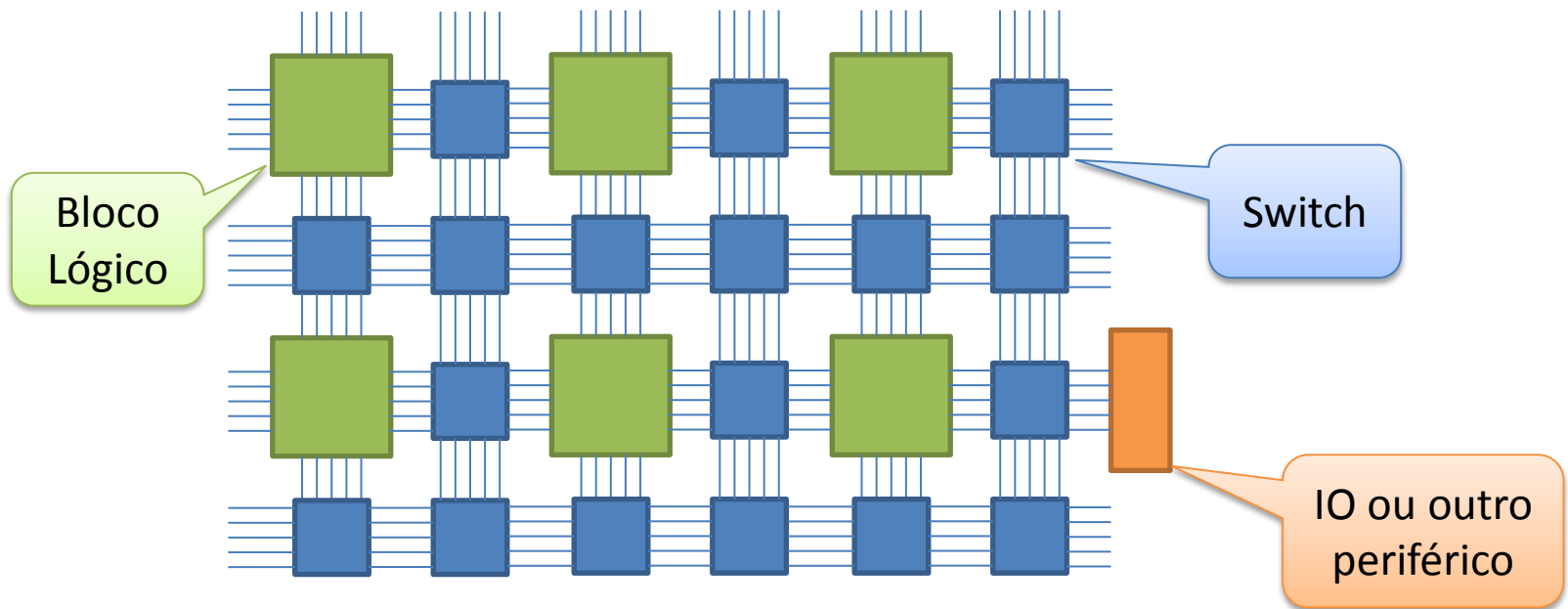
- Multiply-accumulates: $a = a + (bc)$
 - Convolução
 - Cálculo de polinômios
- Algoritmos fundamentais dependem muito do multiply-accumulate
 - Filtros FIR
 - Fast Fourier transform (FFT)
- Instruções para aumentar o paralelismo
 - SIMD: Single instruction, multiple data
 - VLIW: Very long instruction word
- Instruções para ponto flutuante

FPGA (Field-programmable Gate Array)

- Circuito integrado projetado para ser configurado pelo usuário;
- Utiliza-se uma linguagem de descrição de hardware;
- Possuem um vetor de blocos lógicos programáveis e uma hierarquia de interconexões reconfiguráveis, a qual permite interligar os blocos;
 - Pode obter funções complexas ou apenas AND, OR;

FPGAs

- Possuem alto número de blocos lógicos;
- IOs;
- Podem possuir outros blocos para operações complexas;
 - Ex. Comparador analógico, DACs



FPGA

- Podem ser configurados via uma linguagem de descrição de hardware, como VHDL
- Fonte: By Vhdl_signed_adder.png: RevRagnarokderivative work: Bernard Ladenthin - Own work, This file was derived from Vhdl signed_adder.png; CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=29085518>

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.numeric_std.all;
4
5 entity signed_adder is
6   port
7   (
8     aclr : in    std_logic;
9     clk  : in    std_logic;
10    a    : in    std_logic_vector;
11    b    : in    std_logic_vector;
12    q    : out   std_logic_vector
13  );
14 end signed_adder;
15
16 architecture signed_adder_arch of signed_adder is
17   signal q_s : signed(a'high+1 downto 0); -- extra bit wide
18
19 begin -- architecture
20   assert(a'length >= b'length)
21     report "Port A must be the longer vector if different sizes!"
22     severity FAILURE;
23   q <= std_logic_vector(q_s);
24
25   adding_proc:
26   process (aclr, clk)
27   begin
28     if (aclr = '1') then
29       q_s <= (others => '0');
30     elsif rising_edge(clk) then
31       q_s <= ('0'&signed(a)) + ('0'&signed(b));
32     end if; -- clk'd
33   end process;
34
35 end signed_adder_arch;
```