

Trabalho 1 – parte 2

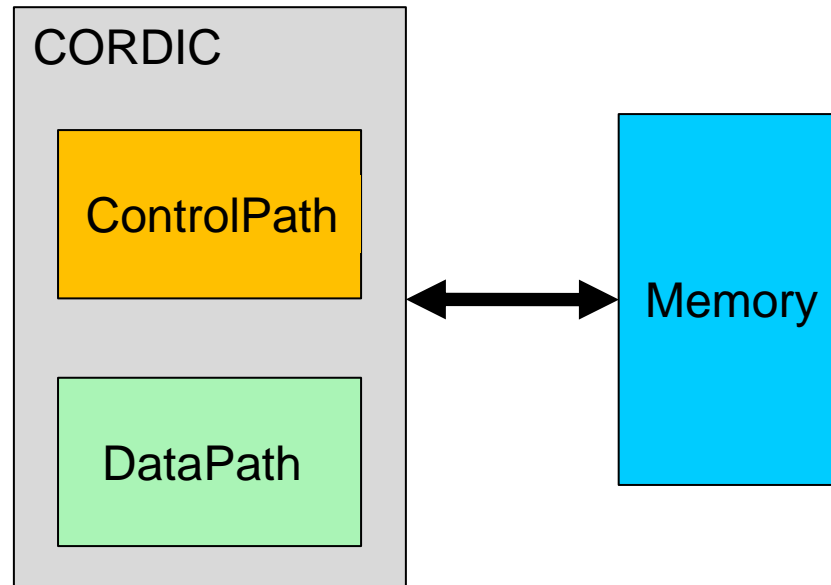
- ❑ Implementar em VHDL duas *architectures* para o processador CORDIC
 1. Comportamental + estrutural (*ControlPath* + *DataPath*)
 - ❑ A descrição VHDL deve ser baseada no projeto implementado na parte 1 do trabalho com a restrição de um único somador para operações aritméticas e comparações
 - ❑ Removam todo o hardware desnecessário do projeto da parte 1 (e.g. demux, barrel-shifter)
 2. Totalmente comportamental
 - ❑ Deve-se explorar o paralelismo a partir da replicação consciente de hardware a fim de aumentar o desempenho em relação ao projeto da parte 1
 - ❑ Espera-se que o tempo de processamento caia para em torno de 1/6 do tempo do projeto da parte 1
-

Trabalho 1 – parte 2

□ *Architecture* Comportamental + estrutural

■ Estrutura da descrição

- 4 entidades: *CORDIC*, *ControlPath*, *DataPath* e *Memory*
- Cada entidade corresponde a um arquivo .vhd de mesmo nome

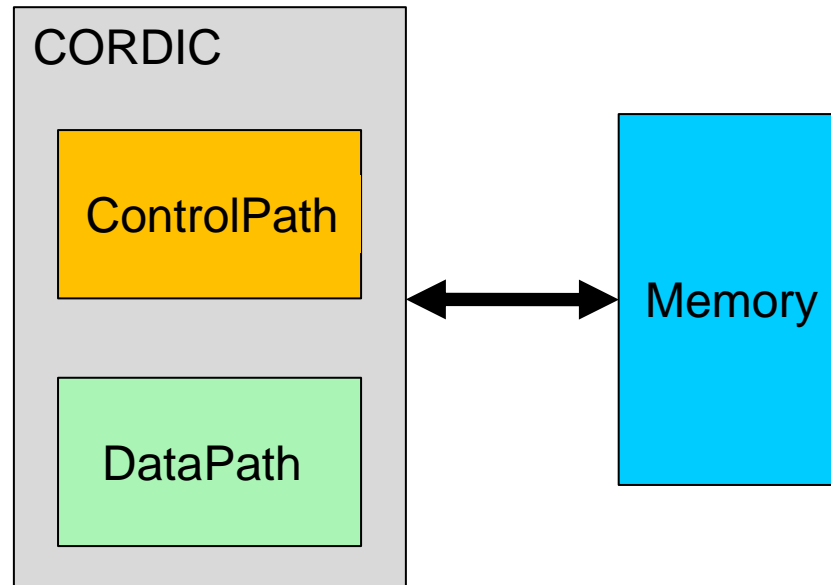


Trabalho 1 – parte 2

□ *Architecture* Comportamental + estrutural

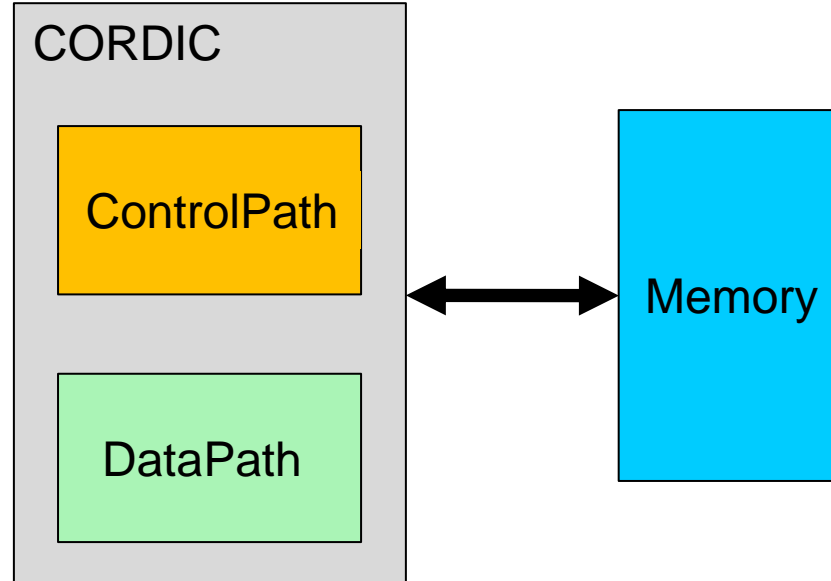
■ Estrutura da descrição

- Criar *package* CORDIC_pkg contendo a definição de uma *record* com todos sinais de controle do *ControlPath* para o *DataPath*



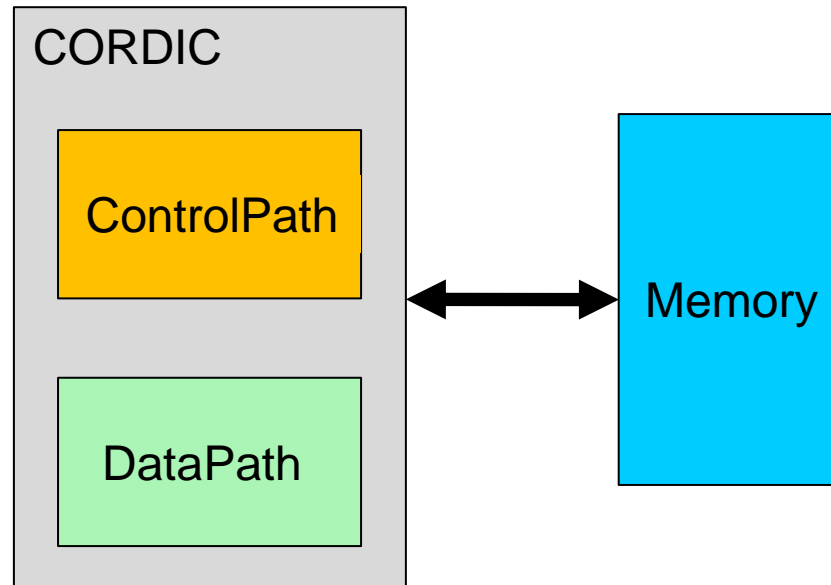
Trabalho 1 – parte 2

- *Architecture* Comportamental + estrutural
 - *DataPath* (DataPath.vhd)
 - Estrutural + comportamental
 - Registradores devem ser instâncias do registrador genérico RegisterNbits (RegisterNbits.vhd)



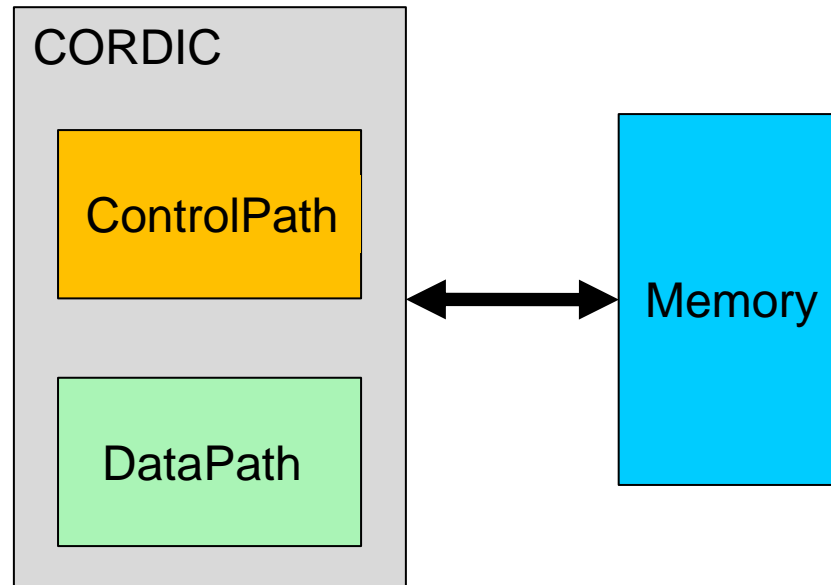
Trabalho 1 – parte 2

- ❑ *Architecture* Comportamental + estrutural
 - *ControlPath* (ControlPath.vhd)
 - ❑ Totalmente comportamental
 - ❑ Registrador de estados + circuitos de saídas e próximo estado



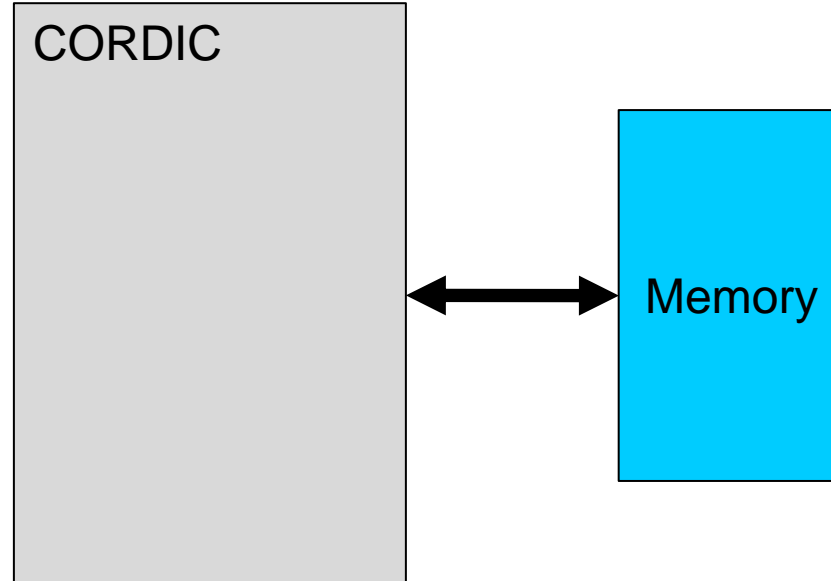
Trabalho 1 – parte 2

- ❑ *Architecture* Comportamental + estrutural
 - CORDIC (CORDIC.vhd)
 - ❑ Totalmente estrutural (ligação entre *DataPath* e *ControlPath*)
 - ❑ Utilizar a *record* definida no CORDIC_pkg na conexão dos blocos



Trabalho 1 – parte 2

- *Architecture* Totalmente comportamental
 - CORDIC (CORDIC.vhd)
 - Totalmente comportamental
 - **Deve ser entregue o esquemático correspondente apenas ao *data path* da descrição implementada**



Trabalho 1 – parte 2

□ Memória (Memory.vhd)

- Deve ter a mesma interface da memória ROM do Logisim com a adição de uma entrada de *clock*
- A leitura deve ser síncrona
- Deve estar habilitada ($sel=1$) apenas quando for utilizada
- Quando $sel=0$ a saída deve ficar em alta impedância (Z)
- Os barramentos devem ser parametrizáveis



Trabalho 1 – parte 2

□ Memória (Memory.vhd)

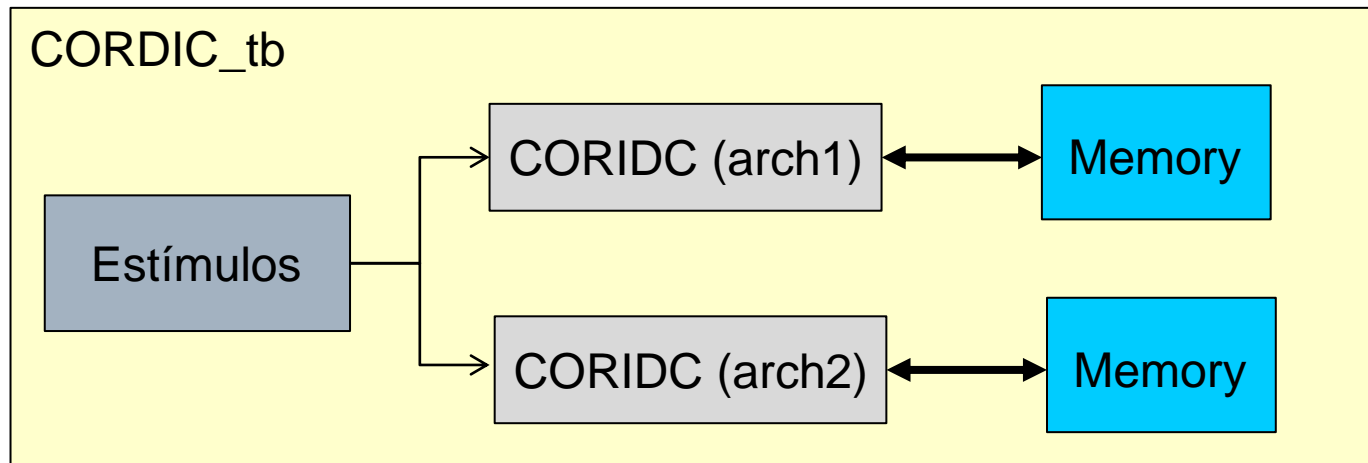
- A memória deve ser inicializada a partir de um arquivo texto
- Será fornecida uma memória de exemplo contendo um procedimento que faz inicialização a partir de um arquivo
- O arquivo deve ter o seguinte formato

Endereços	00	20003042	Dados
	01	AB435555	
	

Trabalho 1 – parte 2

□ *Test bench* (CORDIC_tb.vhd)

- Deve existir um único *test bench* no qual será instanciado as duas descrições do CORDIC e duas memórias
- Os mesmos sinais de estímulos devem ser fornecidos aos dois processadores (*clock, reset, angle...*)
- Na apresentação os dois devem ser simulados simultaneamente a fim de comparar o desempenho



Trabalho 1 – parte 2

- ❑ Será fornecido um processador que implementa o algoritmo *BubbleSort*, o qual apresenta algumas características em comum com o trabalho a ser feito
- ❑ A memória de exemplo é a deste projeto

