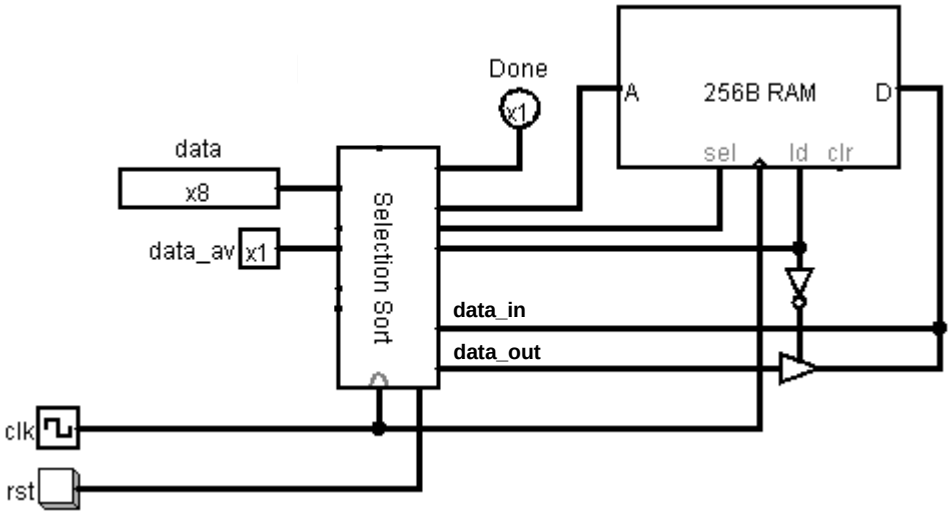
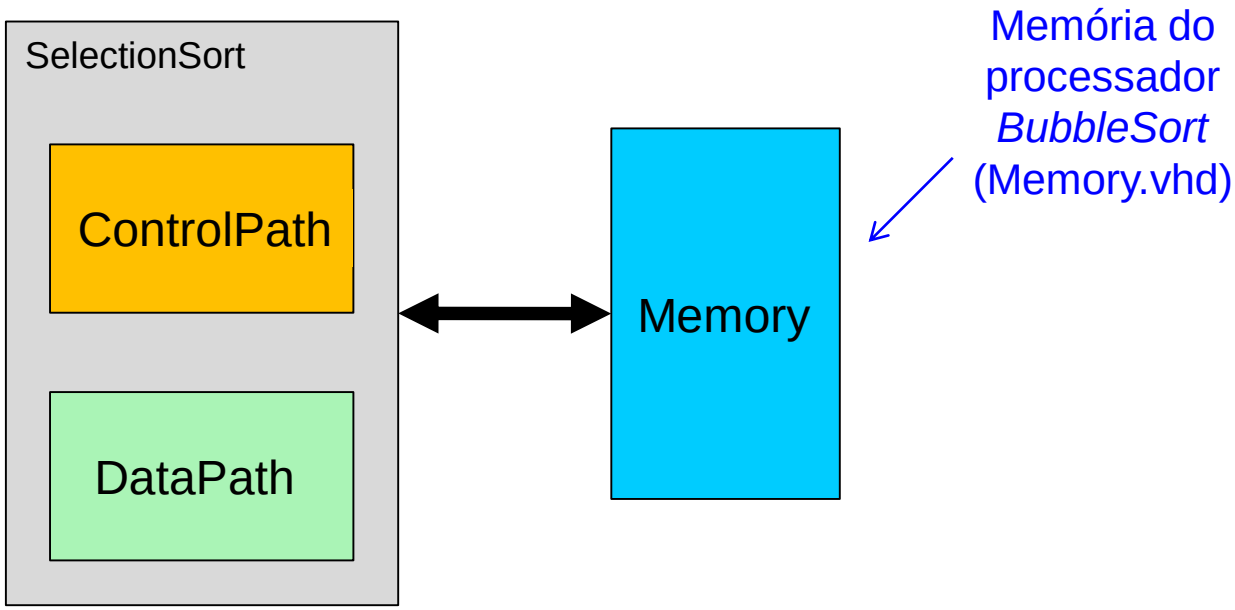
Trabalho 1 – parte 2

- Implementar em VHDL duas *architectures* para o processador *SelectionSort*
  1. Comportamental + estrutural (*ControlPath* + *DataPath*)
     - A descrição VHDL deve ser baseada no projeto implementado na parte 1 do trabalho
  2. Totalmente comportamental
     - A descrição deve ser baseada na FSMD que descreve o comportamento do processador



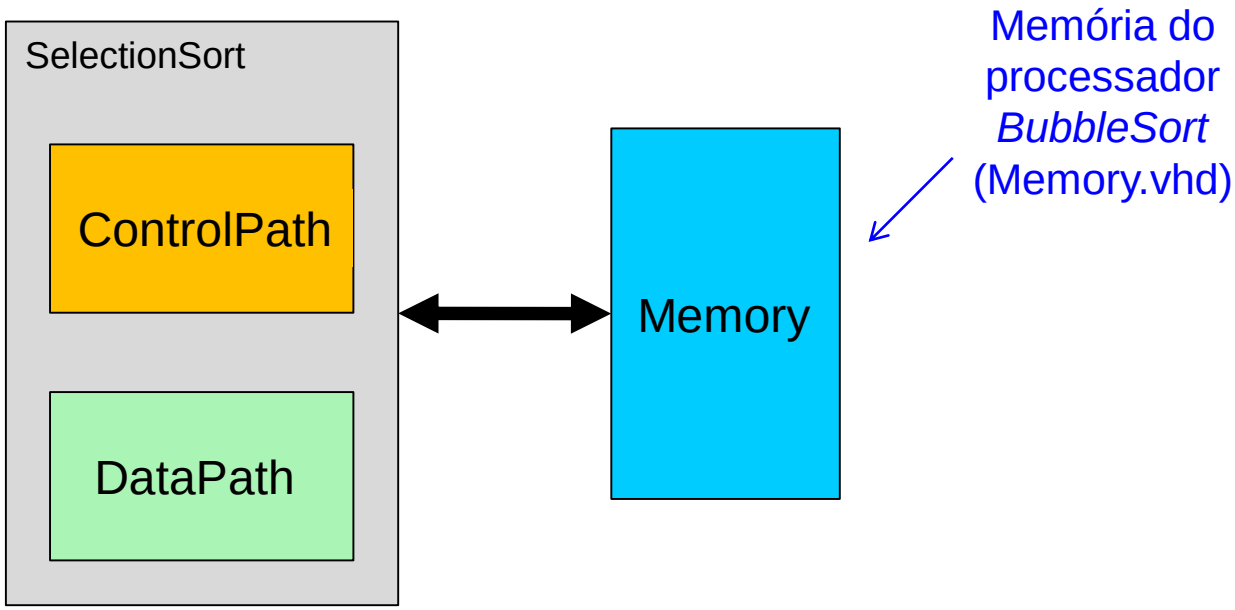
Trabalho 1 – parte 2

- *Architecture* Comportamental + estrutural
  - Estrutura da descrição
    - 4 entidades: *SelectionsSort*, *ControlPath*, *DataPath* e *Memory*
    - Cada entidade corresponde a um arquivo .vhd de mesmo nome



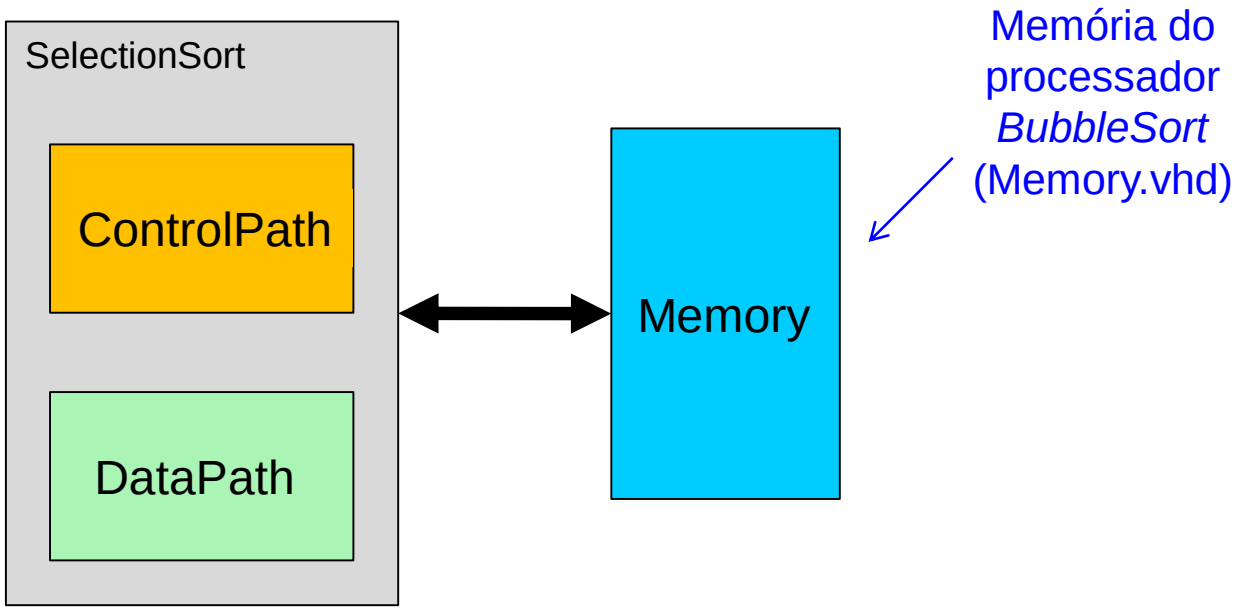
Trabalho 1 – parte 2

- *Architecture* Comportamental + estrutural
  - Estrutura da descrição
    - Criar *package* SelectionSort_pkg contendo a definição de uma *record* com todos sinais de controle do *ControlPath* para o *DataPath*



Trabalho 1 – parte 2

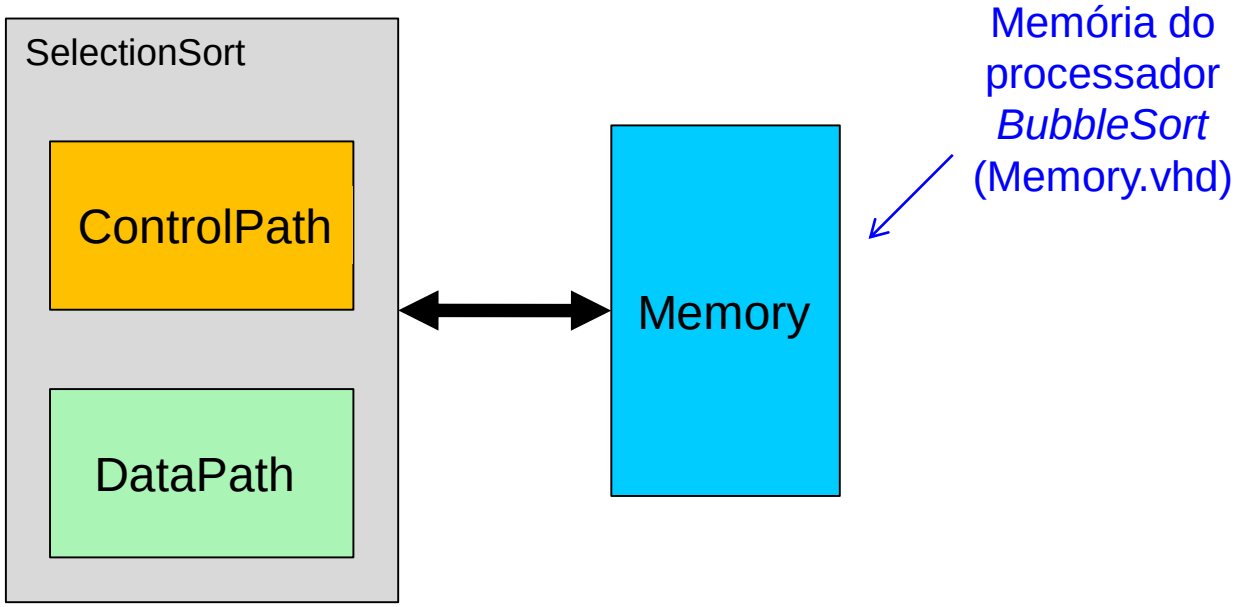
- *Architecture* Comportamental + estrutural
  - *DataPath* (DataPath.vhd)
    - Estrutural + comportamental
    - Registradores devem ser instâncias do registrador genérico RegisterNbits (RegisterNbits.vhd)



Trabalho 1 – parte 2

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

Dica:
- Começar pelo *ControlPath*
- Criar um *test bench* para testar o *ControlPath*

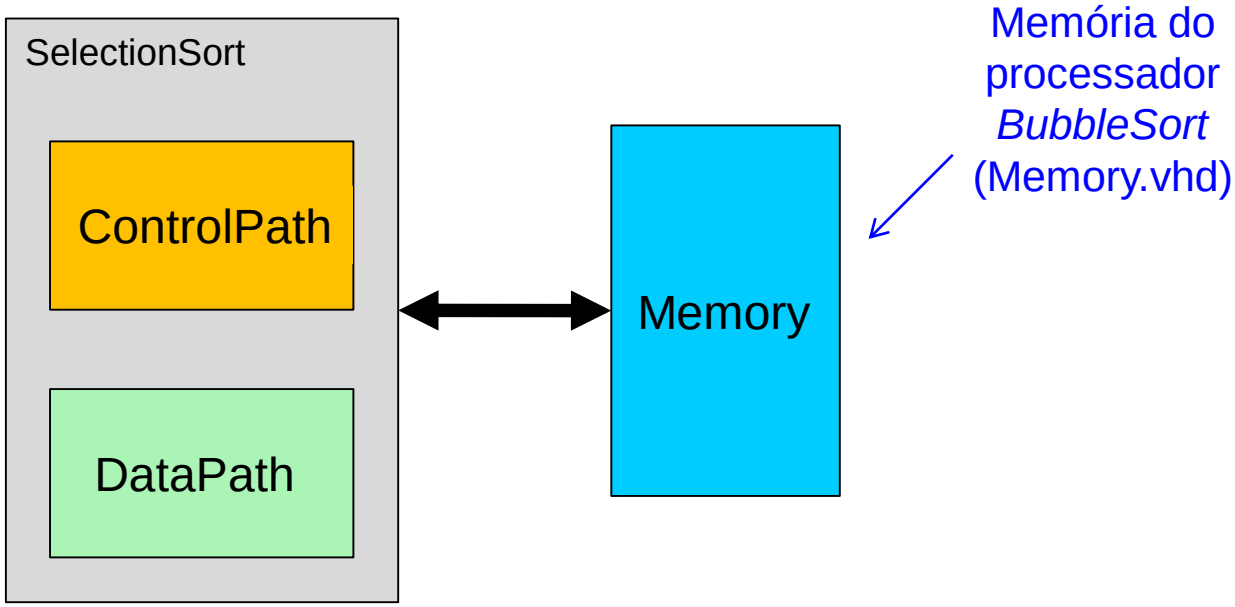


Trabalho 1 – parte 2

- *Architecture* Comportamental + estrutural
  - *SelectionSort* (SelectionSort.vhd)
    - Estrutural (ligação entre *DataPath* e *ControlPath*)
    - Utilizar a *record* definida no SelectionSort_pkg na conexão dos blocos

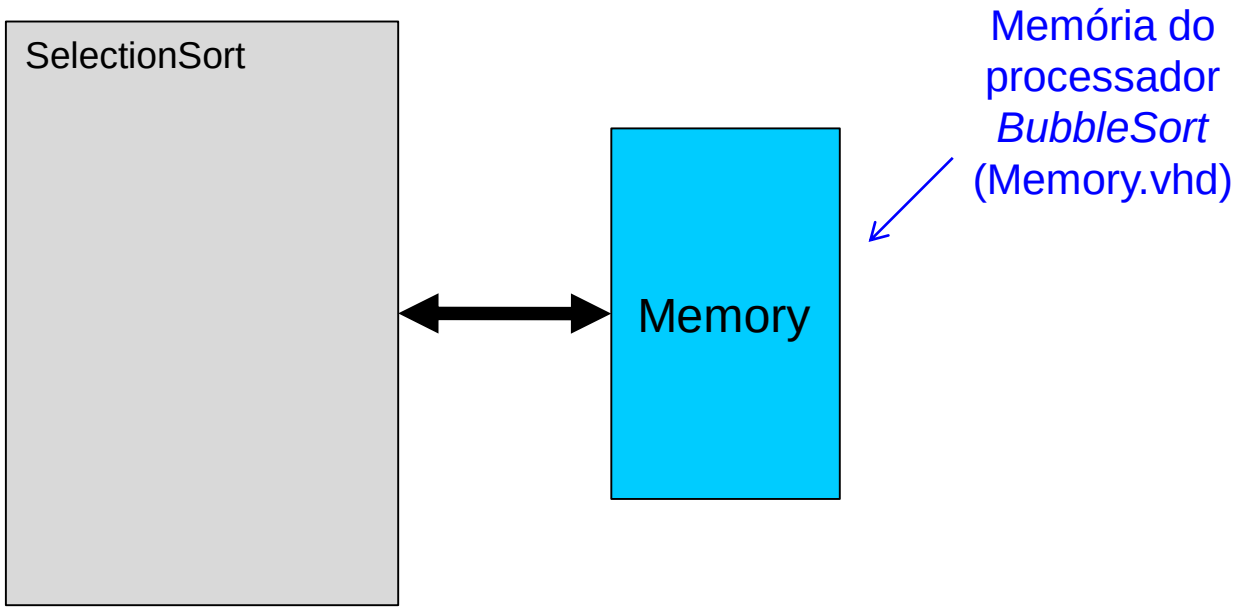
Dica fundamental:

- NÃO DEIXEM PARA TESTAR (*test bench*) QUANDO ESTIVER TUDO DESCRITO EM VHDL!!!



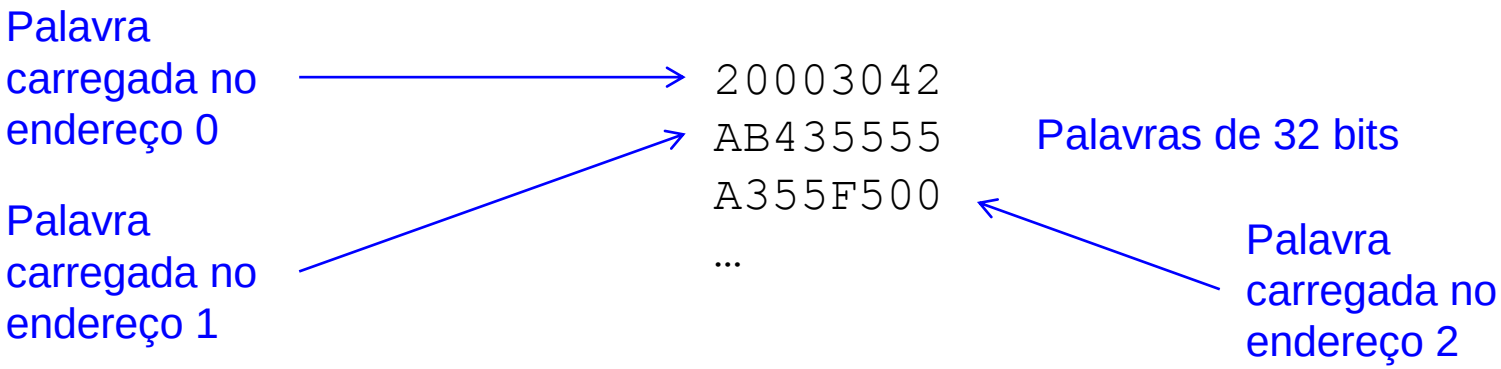
Trabalho 1 – parte 2

- *Architecture* totalmente comportamental
  - SelectionSort (SelectionSort.vhd)
    - 2 entidades: *SelectionSort* e *Memory*
    - **Deve ser entregue o diagrama da FSMD impresso**
    - **Esta descrição deve ter exatamente o mesmo tempo de execução da primeira *architecture***



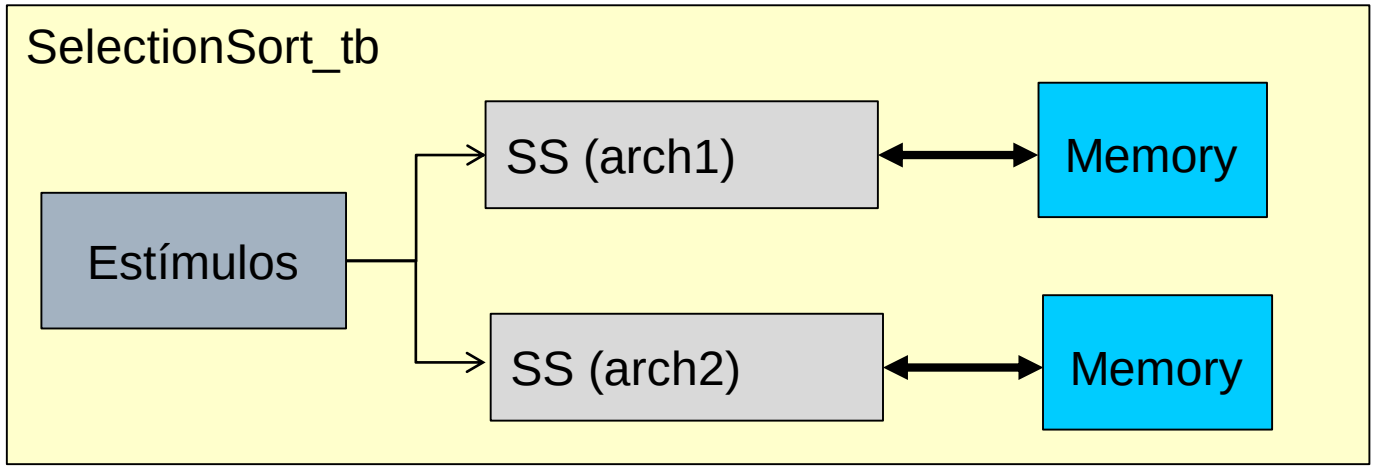
Trabalho 1 – parte 2

- Memória (Memory.vhd)
  - A memória deve ser inicializada a partir de um arquivo texto (parâmetro *imageFileName*)
  - O arquivo deve conter em cada linha um valor hexadecimal. A quantidade de dígitos de cada valor depende da largura da palavra da memória (parâmetro DATA_WIDTH)
  - Cada linha do arquivo corresponde a um endereço da memória
  - Exemplo: DATA_WIDTH = 32



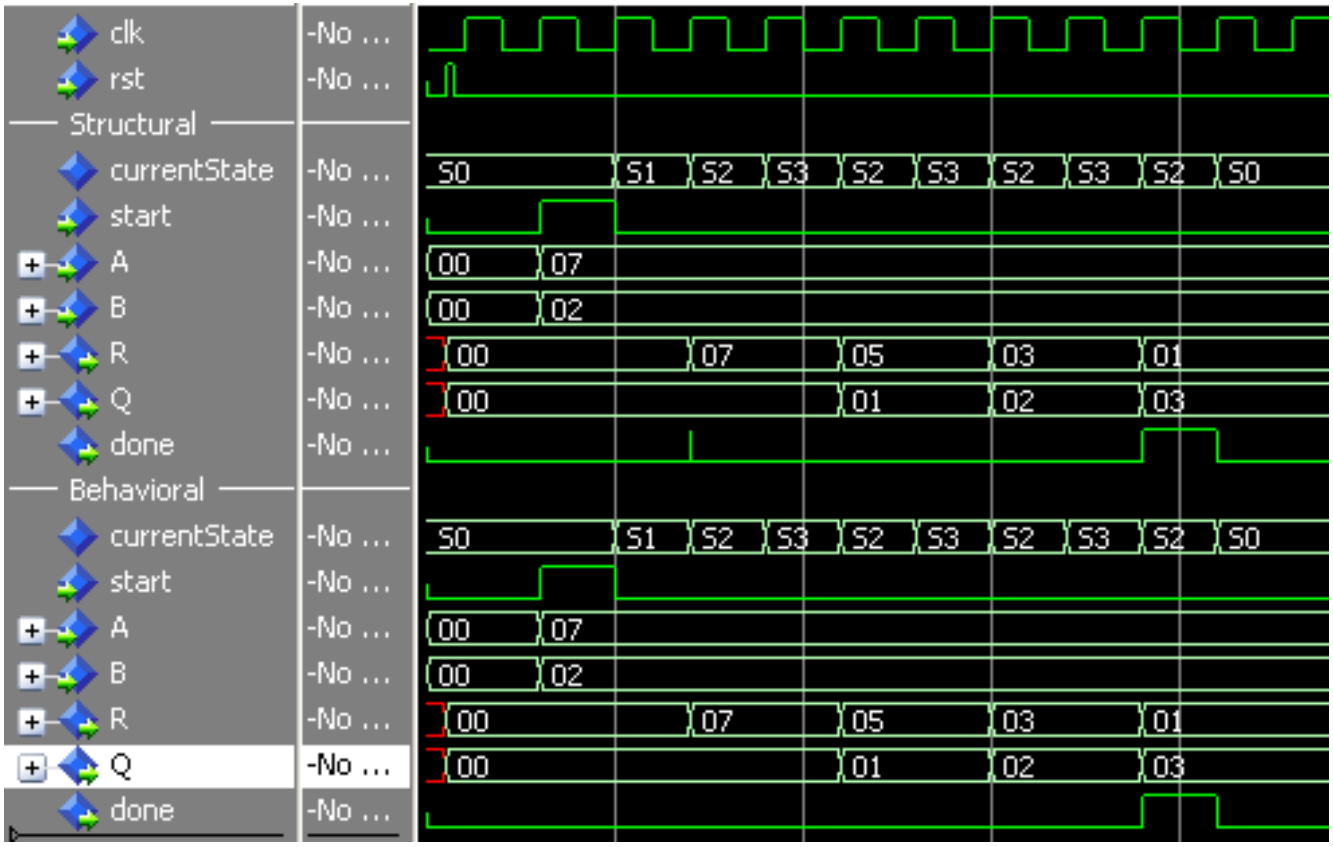
Trabalho 1 – parte 2

- *Test bench* (SelectionSort_tb.vhd)
  - Deve existir um único *test bench* no qual será instanciado as duas descrições do *SelectionSort* e duas memórias
  - Os mesmos sinais de estímulos devem ser fornecidos aos dois processadores (*clock, reset, data, data_av)*
  - Na apresentação, os dois devem ser simulados simultaneamente a fim de verificar exatamente o mesmo comportamento ciclo-a-cilco e o mesmo tempo de execução



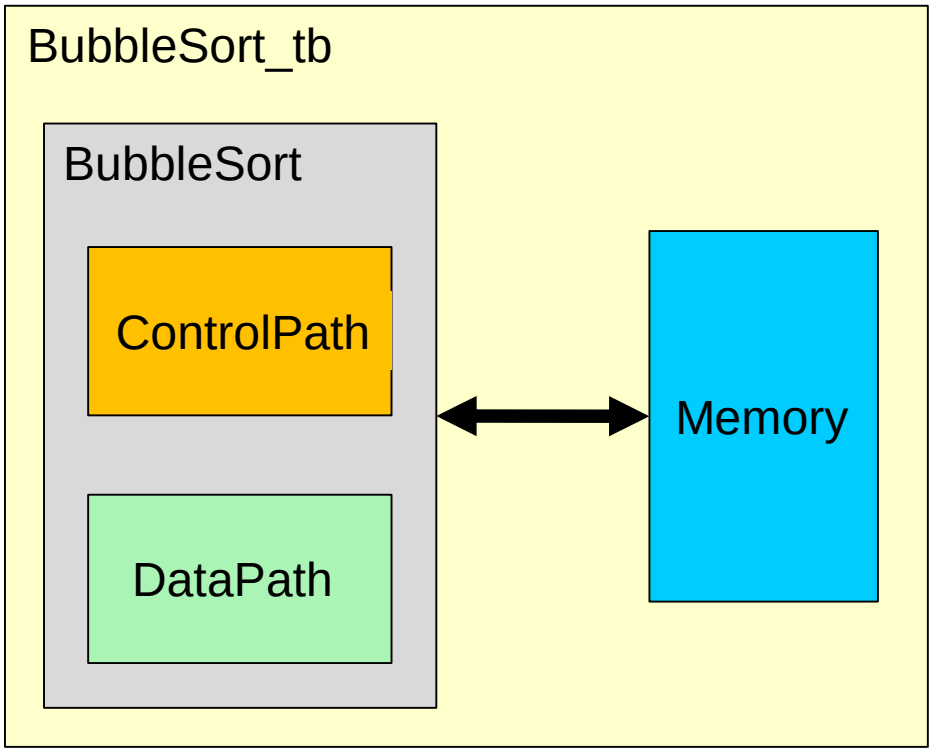
VHDL

- Simulação simultânea de duas *architectures*
  - Divider
    - Exemplo: 7/2 = 3, Resto = 1



Trabalho 1 – parte 2

- Será fornecido um processador que implementa o algoritmo *BubbleSort*, o qual segue o mesmo formato do trabalho a ser feito



Trabalho 1 – parte 2

- Entrega dia 23/6 (todos grupos)
  - Em relação à descrição totalmente comportamental
    - Entregar grafo da FSMD impresso
  - Arquivos VHDL submetidos via *moodle* por um integrante do grupo
- Apresentação 23/6 e 26/6
- Para a parte 2 do trabalho, os mesmos grupos deverão ser mantidos