

Atividade sobre projeto em RTL com BO+BC

Prof. Ilan Sousa Correa

Universidade Federal do Pará (UFPA)

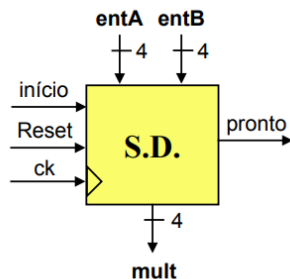
Instituto de Tecnologia (ITEC)

Faculdade de Eng. da Computação e Telecomunicações (FCT)

Problema 1 – multiplicação por somas sucessivas

Considere que deve-se implementar um sistema digital que implementa o algoritmo abaixo a esquerda. Deve-se utilizar o modelo BO+BC para embarcar este algoritmo no bloco SD que possui as interfaces mostradas abaixo a direita. Por questões de custo, deve ser utilizado um único somador/subtrator para realizar as operações $P+B$ e $A-1$, ou seja, este somador subtrator será compartilhado para a realização dessas duas operações, logo cada uma delas deverá ser realizada em um ciclo de clock diferente.

```
início
pronto ← 0;
A ← entA;
B ← entB;
P ← 0;
Se B ≠ 0 então
Enquanto A ≠ 0 faça
    início
        P ← P + B;
        A ← A - 1;
    fim
mult ← P;
pronto ← 1;
fim
```



Problema 2 – medidor de tempo de reação

Desenvolva um sistema digital medidor de tempo de reação que mede o tempo decorrido entre o acendimento de uma lâmpada e o apertar de um botão por uma pessoa. O medido tem três entradas, uma entrada **clk** de clock, uma entrada **rst** de reset e um botão de entrada **B**. Tem três saídas, uma saída **len** de habilitação da lâmpada, uma saída **rtempo** de tempo de reação de dez bits e uma saída **lento** para indicar que o usuário não foi rápido o suficiente. Durante o reset, o medidor de tempo de reação espera por 10 segundos antes de acender a lâmpada fazendo **len=1**. A seguir, o medidor de tempo de reação mede o intervalo de tempo decorrido em milissegundos até o usuário pressionar o botão B, fornecendo o tempo como um número binário de 12 bits na saída **rtempo**. Se o usuário não pressionar o botão dentro de 2 segundos, o medidor irá ativar a saída **lento** tornando-a 1 e colocando 2 segundos em **rtempo**. Assuma que a entrada de clock tem uma frequência de 2,5 kHz. O tempo de reação deve ser fornecido na saída **rtempo** em segundos.

Problema 3 – máquina de vendas melhorada

Crie um sistema digital para melhorar a máquina de vendas da aula anterior. A máquina deve permitir a escolha de dois tipos de refrigerante e também fornecer troco ao consumidor. Um detector de moedas fornece ao circuito um sinal **m** de um bit que torna-se 1 durante 1 ciclo de clock quando uma moeda é detectada, e também a quantia **v** de oito bits, com o valor da moeda em centavos. Duas entradas **r1** e **r2** de oito bits indicam o custo dos dois produtos. A escolha do produto é controlada por dois botões **b1** e **b2** que, quando pressionados, ficam em 1 durante um ciclo de clock. Se moedas suficientes foram inseridas, então o sistema coloca um dos bits de saída **f1** ou **f2** em 1 durante um ciclo de clock para fornecer o produto escolhido. Se for necessário troco, o sistema colocará um bit **nt** em 1 durante um ciclo de clock e também indicará na saída **vt** de oito bits o valor necessário do troco.

Problema 4 – somador de números positivos em memória

Crie um sistema digital que deve somar todos os números que estão armazenados em uma memória de acordo com o seguinte critério. Se a entrada **P** for 1, deve-se somar os números positivos, se for 0, deve-se somar os números negativos. A memória tem espaço para 100 números de 32 bits, e possui uma entrada **end** que indica o endereço a ser lido, e uma saída **dado** para fornecer o dado no endereço no mesmo instante que **end** altera o seu valor (sem atraso). O sistema digital deve checar todas as posições da memória e somar de acordo com o critério fornecido. Ao finalizar, o sistema fornece o resultado em uma saída **soma**. Por questões de custo, deve ser utilizado um único somador para calcular o endereço e a soma dos valores, ou seja, este somador será compartilhado para a realização dessas duas operações, logo cada uma delas deverá ser realizada em um ciclo de clock diferente.

Orientações gerais

Usando o método de projeto RTL e a abordagem BO+BC, implemente em VHDL sistemas digitais para solução dos problemas anteriores.

- Em equipe de até três alunos
- Sorteio aleatório dos problemas
- Apresentação no dia 03/11
- 10 minutos por equipe
- Entrega via SIGAA em PDF da apresentação incluindo capturas das simulações e arquivos VHDL gerados. Um dos membros da equipe deve entregar. O arquivo PDF deve incluir nome, sobrenome e matrícula dos membros da equipe.
- Simular usando um testbench e mostrar como as interações com componentes externos ao sistema desenvolvido foram simuladas.

Orientações gerais

Explicar o funcionamento da solução.

- Esboçar bloco operativo: somadores, comparadores, multiplexadores, registradores. Explicar nomes dos sinais. Utilizar, se necessário, o RTL Viewer
- Esboçar bloco de controle: máquinas de estados e seus sinais de entrada e de saída. Utilizar se necessário o RTL Viewer/State Machine Viewer do Quartus

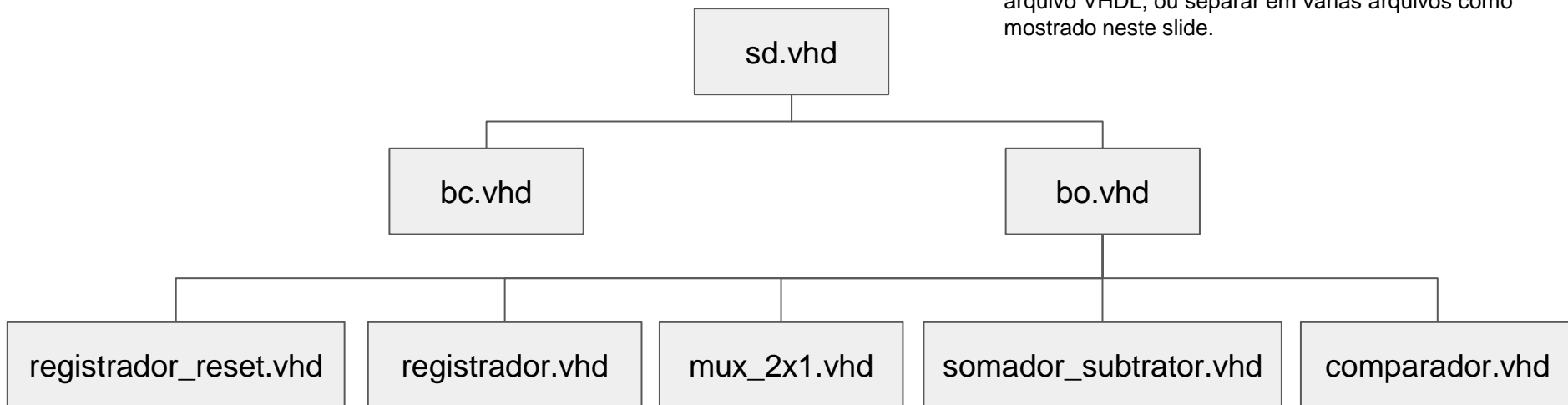
Implementar em VHDL

- Bloco operativo, bloco de controle e testbench para avaliar a funcionalidade.
- Esboçar formas de onda para auxiliar a explicação
- Utilizar simulação no Questasim (ou outra ferramenta) com as devidas explicações.

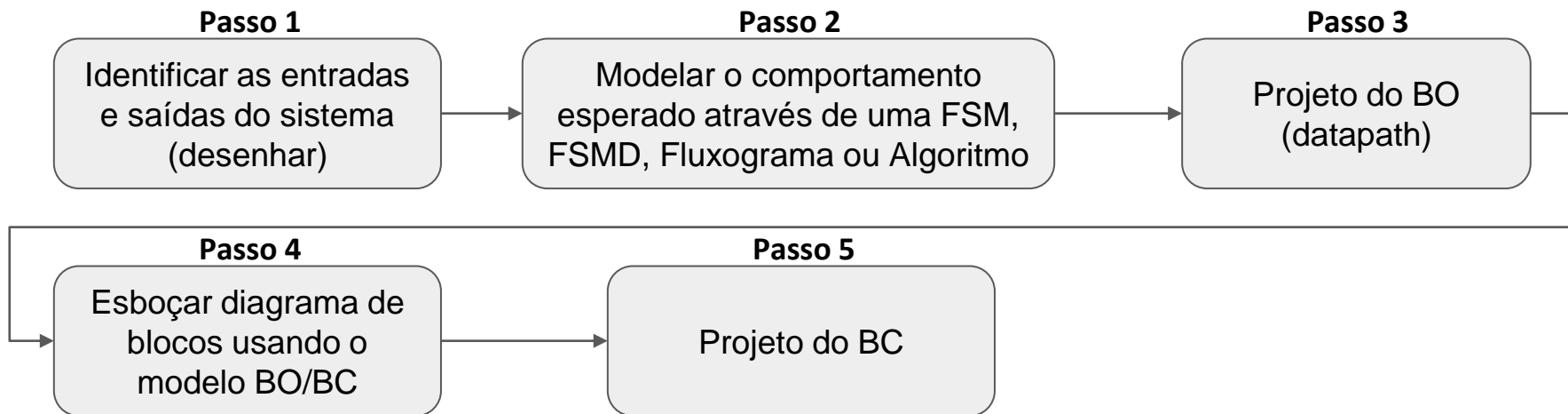
Exemplo de possível hierarquia de arquivos

Exemplo de hierarquia de arquivos

De acordo com o que foi exposto na aula, a equipe pode implementar várias funcionalidade em um único arquivo VHDL, ou separar em várias arquivos como mostrado neste slide.



Revisão dos passos de projeto estudados

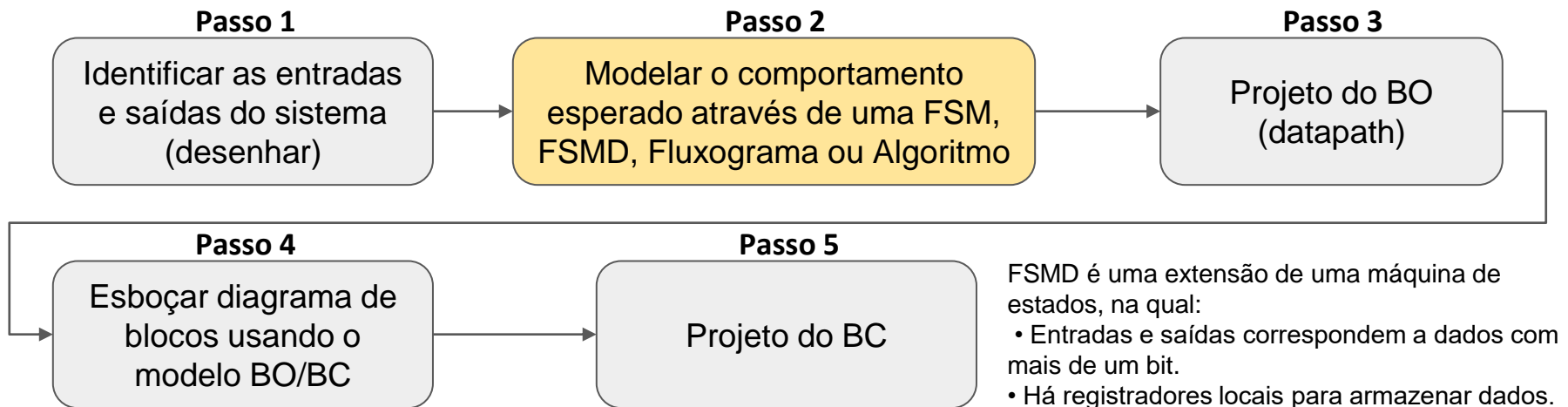


FSM \Rightarrow Finite State Machine \Rightarrow Máquina de estados finitos

FSMD \Rightarrow Finite State Machine with Data \Rightarrow Máquina de estados finitos com dados

UFPA/ITEC/FCT

Revisão dos passos de projeto estudados



FSMD é uma extensão de uma máquina de estados, na qual:

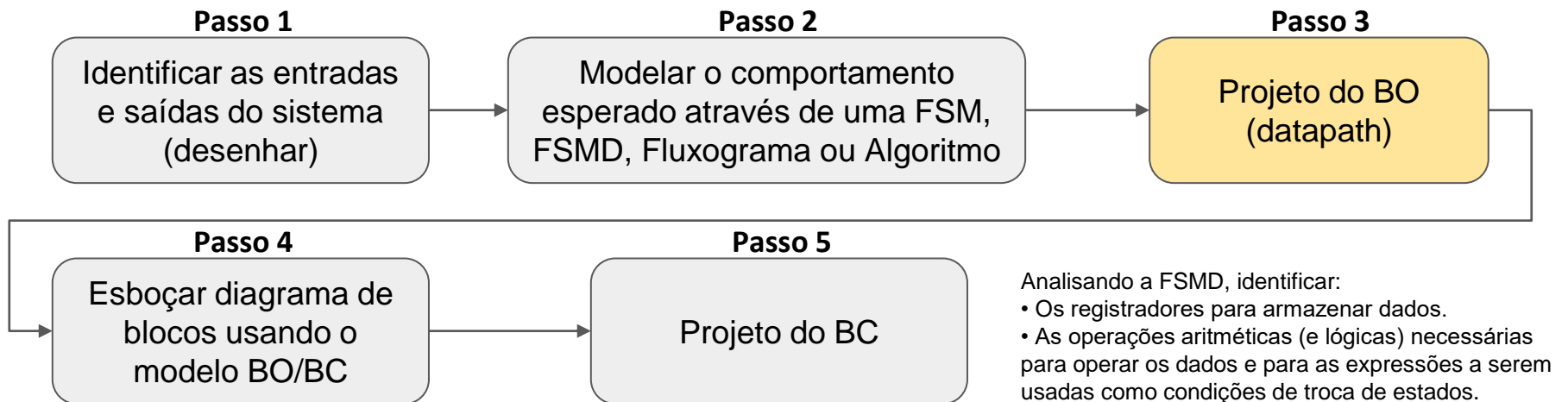
- Entradas e saídas correspondem a dados com mais de um bit.
- Há registradores locais para armazenar dados.
- Ações e condições podem envolver equações e expressões aritméticas (ao invés de apenas equações e expressões Booleanas).

FSM ⇒ Finite State Machine ⇒ Máquina de estados finitos

FSMD ⇒ Finite State Machine with Data ⇒ Máquina de estados finitos com dados

UFPA/ITEC/FCT

Revisão dos passos de projeto estudados



Analisando a FSMD, identificar:

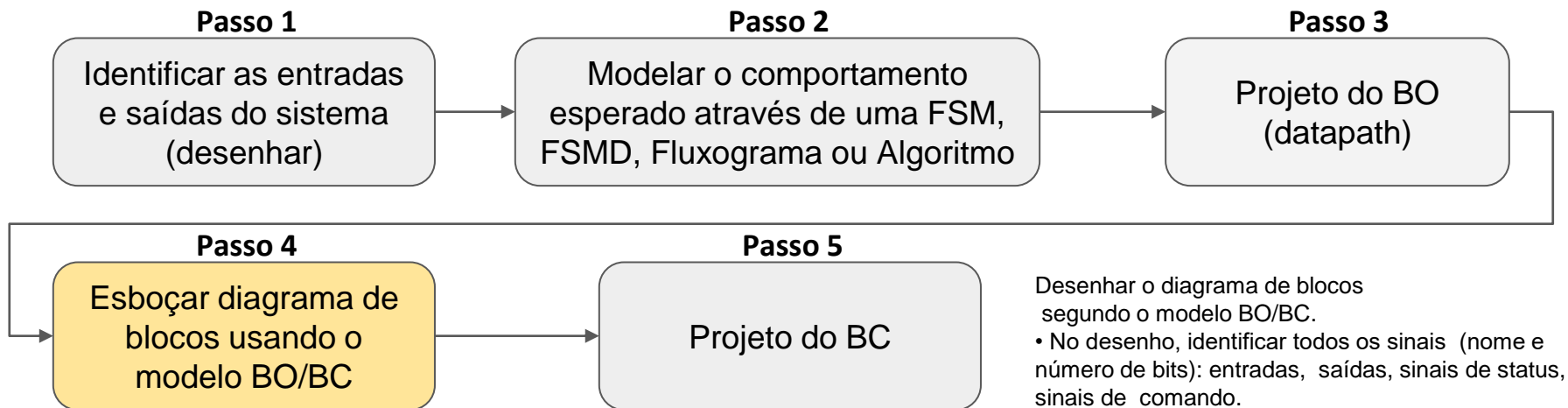
- Os registradores para armazenar dados.
- As operações aritméticas (e lógicas) necessárias para operar os dados e para as expressões a serem usadas como condições de troca de estados.
- Selecionar os componentes do nível RT para implementar, conforme identificado no passo anterior.
- Conectar os componentes do nível RT selecionados no passo anterior.

FSM \Rightarrow Finite State Machine \Rightarrow Máquina de estados finitos

FSMD \Rightarrow Finite State Machine with Data \Rightarrow Máquina de estados finitos com dados

UFPA/ITEC/FCT

Revisão dos passos de projeto estudados

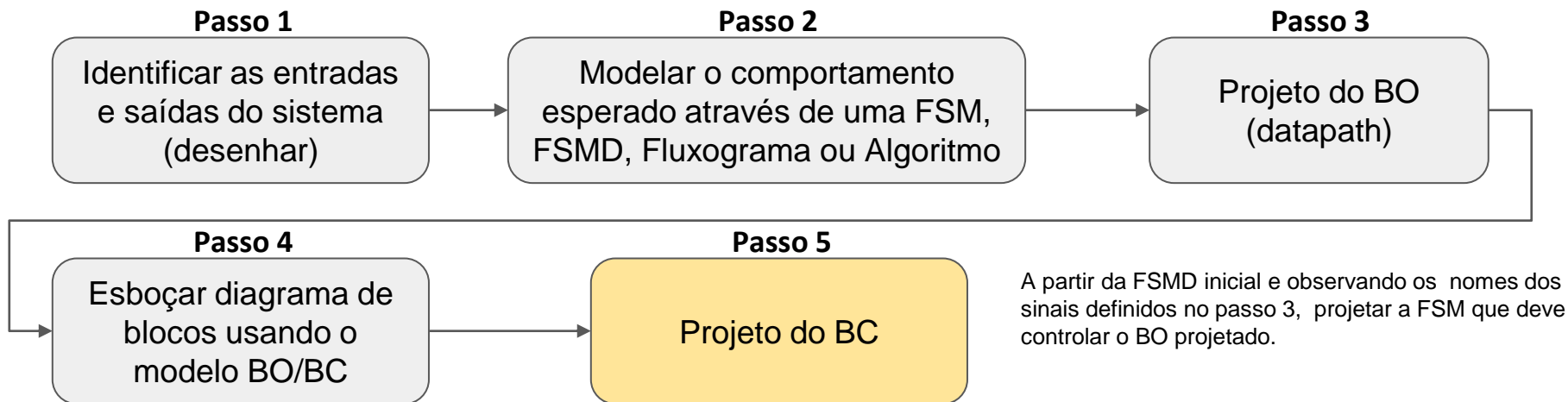


FSM ⇒ Finite State Machine ⇒ Máquina de estados finitos

FSMD ⇒ Finite State Machine with Data ⇒ Máquina de estados finitos com dados

UFPA/ITEC/FCT

Revisão dos passos de projeto estudados



FSM ⇒ Finite State Machine ⇒ Máquina de estados finitos

FSMD ⇒ Finite State Machine with Data ⇒ Máquina de estados finitos com dados

UFPA/ITEC/FCT

Dúvidas?