Architecting Speed

By Prof. Vanderlei Bonato

Estudo de caso – cálculo da potência de 3

- Código sequencial para X^3:

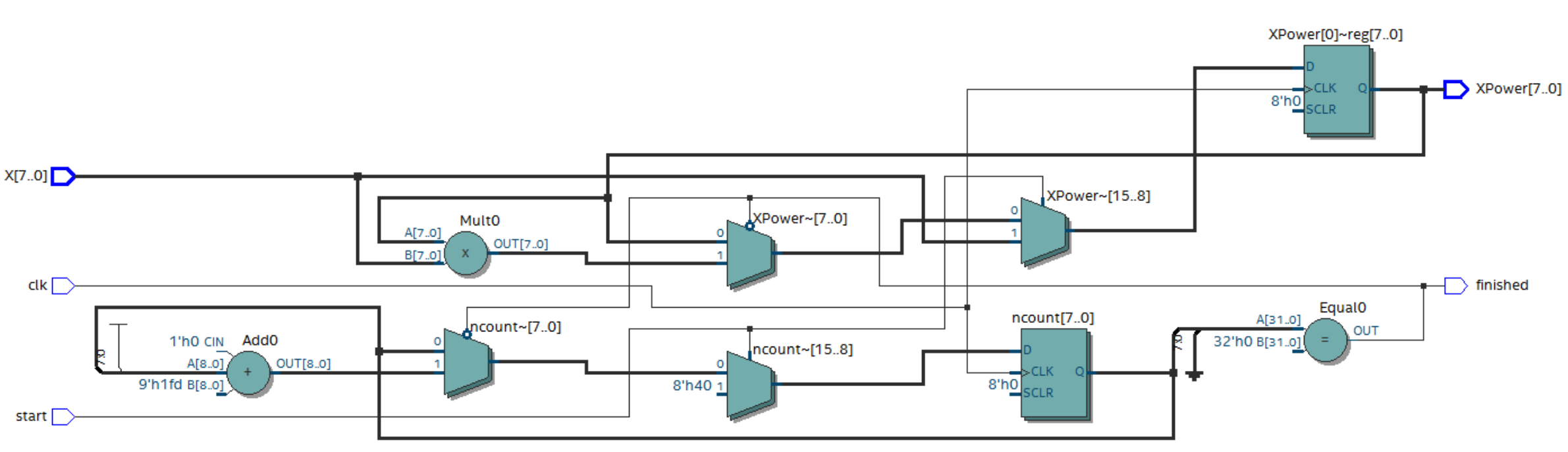
```
X = ?
XPower = 1;
for(i=0; i<3; i++)
  XPower = XPower * X;
```

Cálculo sequencial em hw

```
module XPowerIterative(
output        reg[7:0] XPower,
output                 finished,
input [7:0] X,
input clk, start);

reg [7:0] ncount;
assign finished = (ncount == 0);

always@(posedge clk)begin
        if(start) begin
                XPower <= X;
                ncount <= 2;
        end
        else if(!finished) begin
                ncount <= ncount -1;
                XPower <= XPower * X;
        end
end
endmodule
```

module XPowerIterative

Throughput?
Latency?
Timing?

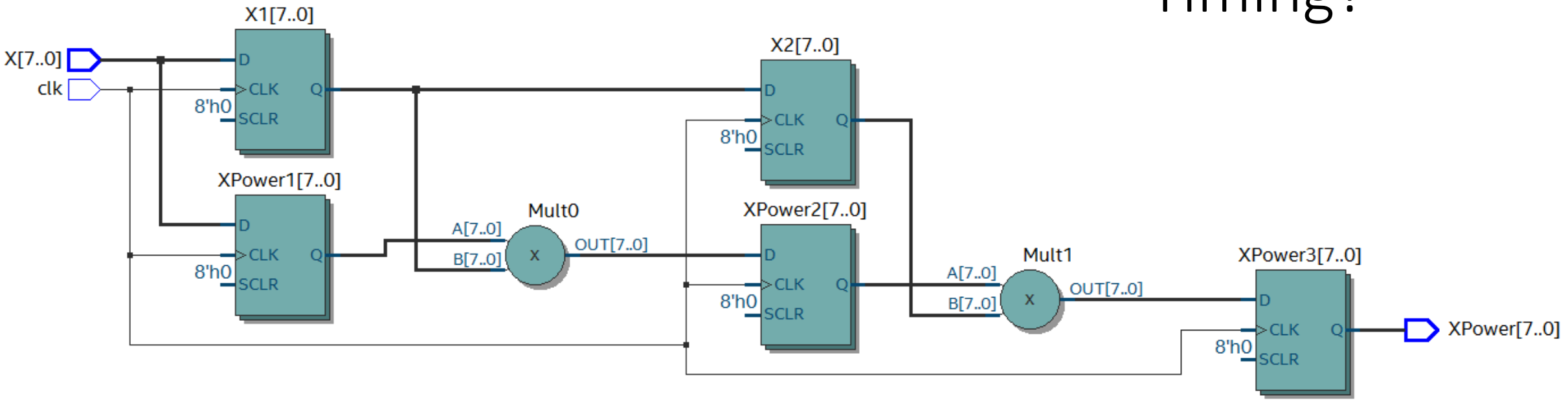


module XPowerPipelined

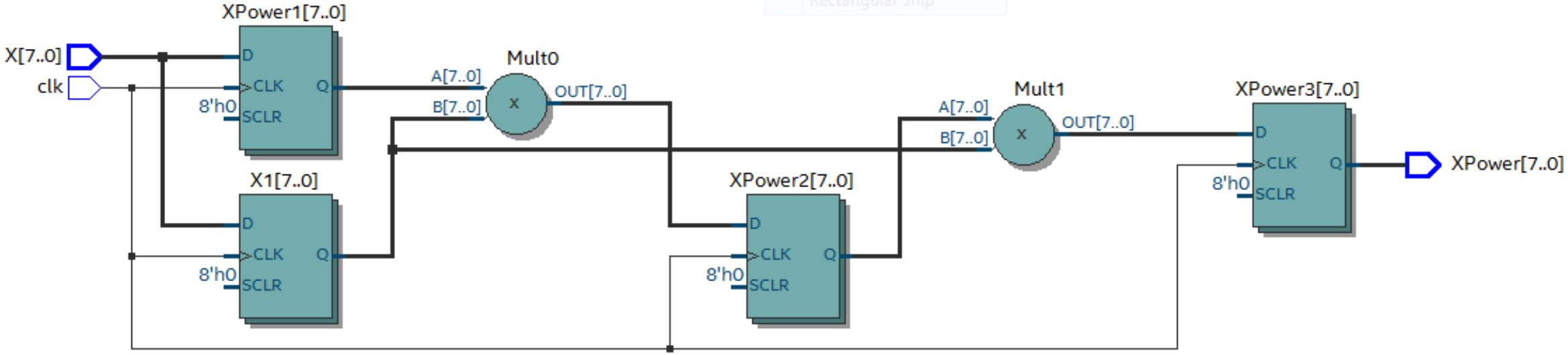
```
module XPowerPipelined(
output [7:0]XPower,
input  [7:0]X,
input clk);

reg [7:0] XPower1, XPower2, XPower3;
reg [7:0] X1, X2;

assign XPower = XPower3;
always@(posedge clk)begin
        //Load pipeline
        X1 <= X;
        XPower1 <=X;

        //Pipeline stage 1
        X2 <= X1;
        XPower2 <= XPower1 * X1;

        //Pipeline stage 2
        XPower3 <= XPower2 * X2;  //qual o efeito no pipeline se usar X1 no lugar de X2?

end
endmodule
```

module XPowerPipelined

Throughput?
Latency?
Timing?



Qual o efeito no pipeline se usar X1 no lugar de X2?



Throughput?
Latency?
Timing?

module combinatorial

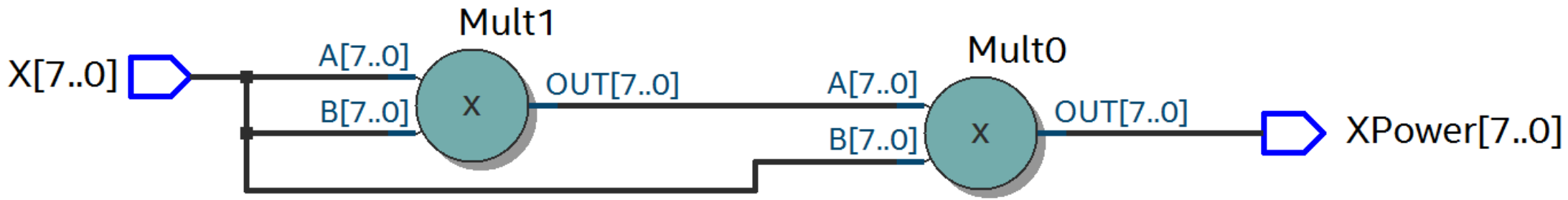
```
module combinatorial(
        output [7:0]XPower,
        input  [7:0]X
        );
        reg [7:0] XPower1, XPower2;
        reg [7:0] X1, X2;
        assign XPower = XPower2 * X2;  //mult 2
        always @* begin
                //Load data
                X1 <= X;
                XPower1 <=X;

        end
        always @* begin
                //mult 1
                X2 <= X1;
                XPower2 <=XPower1 * X1;
        end
endmodule
```

Or

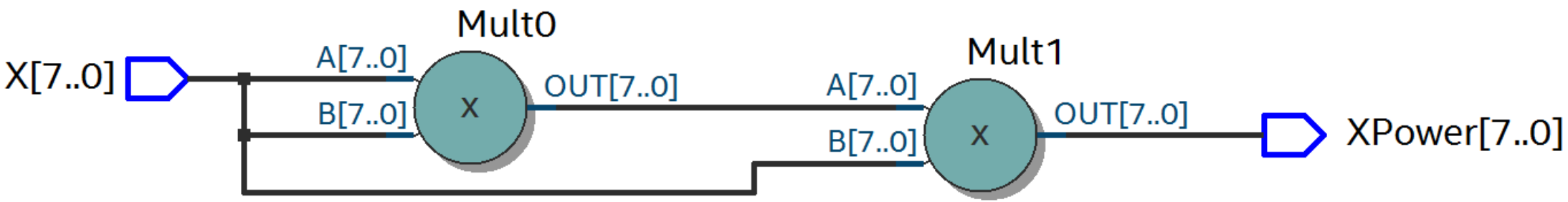
```
module combinatorial(
        output [7:0]XPower,
        input  [7:0]X
        );

        assign XPower = X * X * X; //x2

endmodule
```

module combinatorial

Throughput?
Latency?
Timing?



Or



Exercício 1

- Simular todos esses circuitos
- Sintetizar para obter a frequência de operação
- Verificar para cada circuito
  - Throughput
  - Latency
  - Timing

Exercício 2

- Crie hardware para multiplicação de matriz nas três versões:
  - Iterativa,
  - Pipeline e
  - Combinacional;
- Tamanho da matriz $A_{5,3}$ e $B_{3,4}$;
- Números inteiros de 16 bits cada