
Prototipação de Circuitos **Integrados**

**CONCURSO PÚBLICO PARA PROFESSOR UNIVERSITÁRIO,
EDITAL Nº 02, 10 DE FEVEREIRO DE 2010**

Prova didática

Candidato: Daniel Augusto Ribeiro Chaves

Como configurar o FPGA?

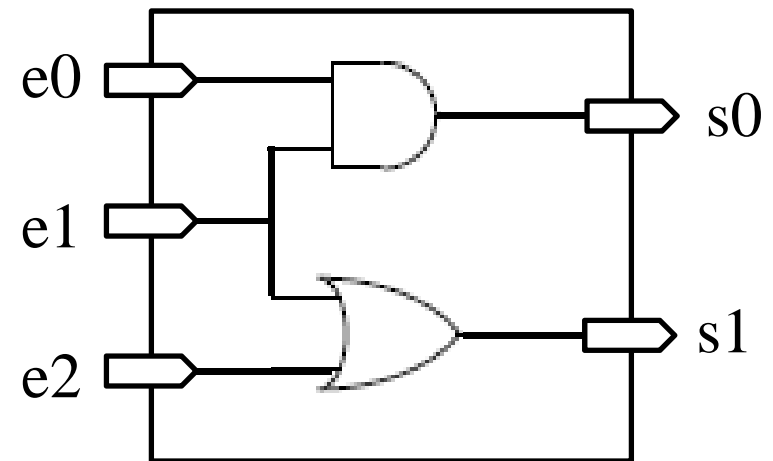
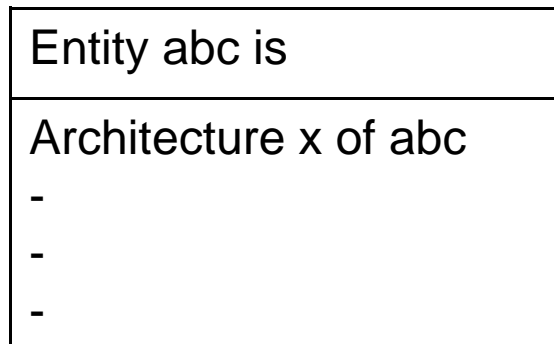
- Imagine o seguinte cenário:
 - Um sistema com dezenas de milhares de portas lógicas.
 - Projete o sistema e encontre uma forma de implementar no FPGA se preocupando com interligações de blocos lógicos, LUTs etc.
- Muito complexo!
- Este processo deve ser feito de forma automática:
 - Projetista descreve o sistema e um “compilador” sintetiza e encontra uma forma de colocar no FPGA.

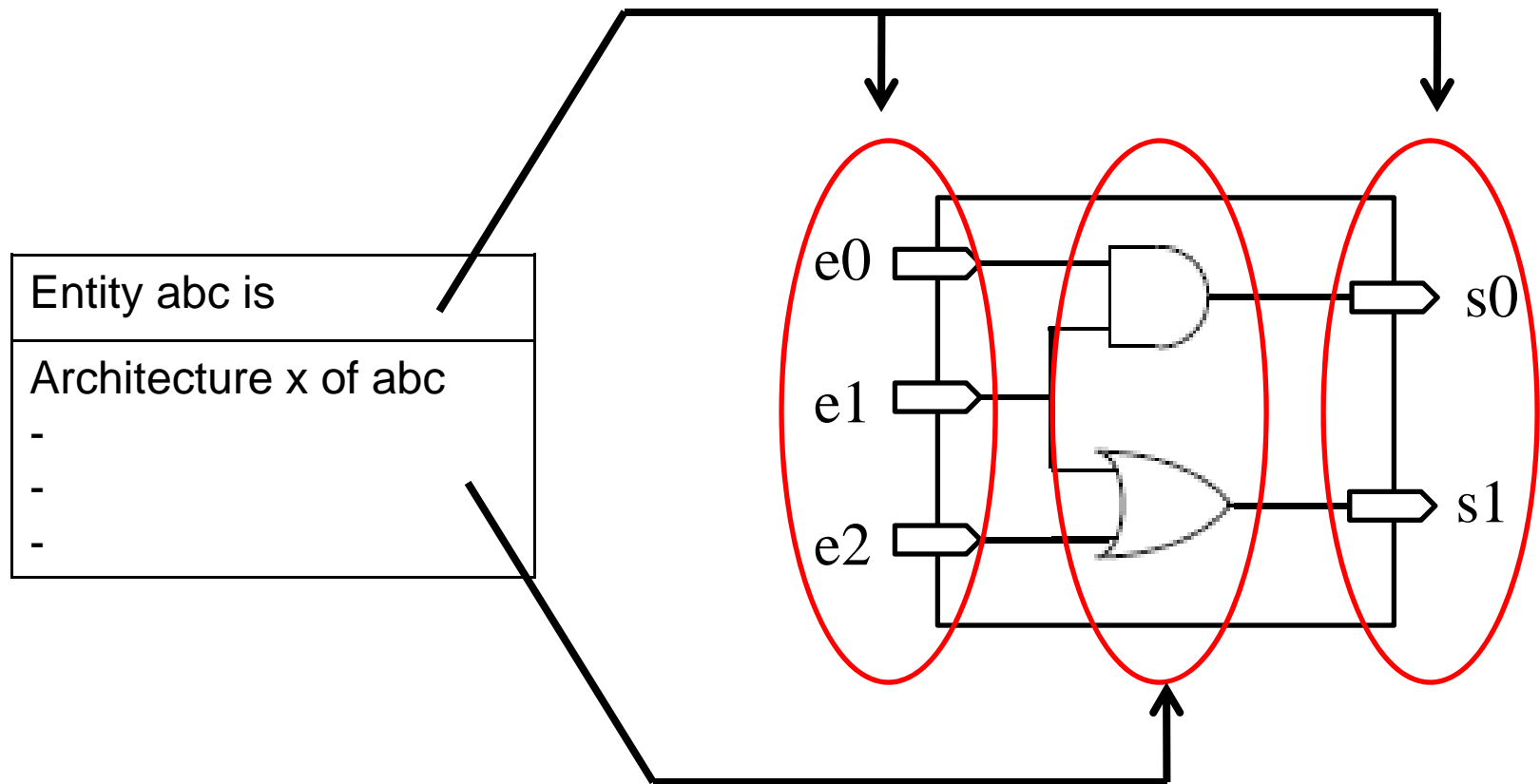
- Uma alternativa à entrada esquemática de um circuito digital.
- O projetista cria um arquivo de texto, seguindo certo conjunto de regras, conhecido como sintaxe da linguagem.
- Uma ferramenta CAD gera o circuito digital de forma automática a partir da descrição.

- Linguagens:
 - Hardware Description Language (HDL)
 - Altera Hardware Description Language (AHDL)
 - VHSIC Hardware Description Language (VHDL)
 - Verilog
- As duas últimas são padronizadas pelo IEEE.

- A linguagem VHDL foi originalmente desenvolvida por empresas contratadas pelo governo.
- Padronizada pelo IEEE em 1987.
- Declarações concorrentes (por *default*) ou seqüencial.
- Ordem dos comandos é irrelevante.
- Não é sensível ao caso.
- Podem ser feitas sub-rotinas e subprogramas.
- Nem todas as funções da linguagem podem ser implementadas em software
 - Ex: Flip-Flop sensível a borda de subida e descida ao mesmo tempo.

- Entidade – Define a interface entre uma entidade e o mundo exterior (Ex. entradas e saídas).
- Arquitetura – Define as relações entre entrada e saída.





Exemplo – VHDL

Entity portas IS

```
PORT(a, b, c : IN BIT
```

```
      x, y : OUT BIT);
```

```
END Portas
```

```
ARCHITECTURE teste OF portas IS
```

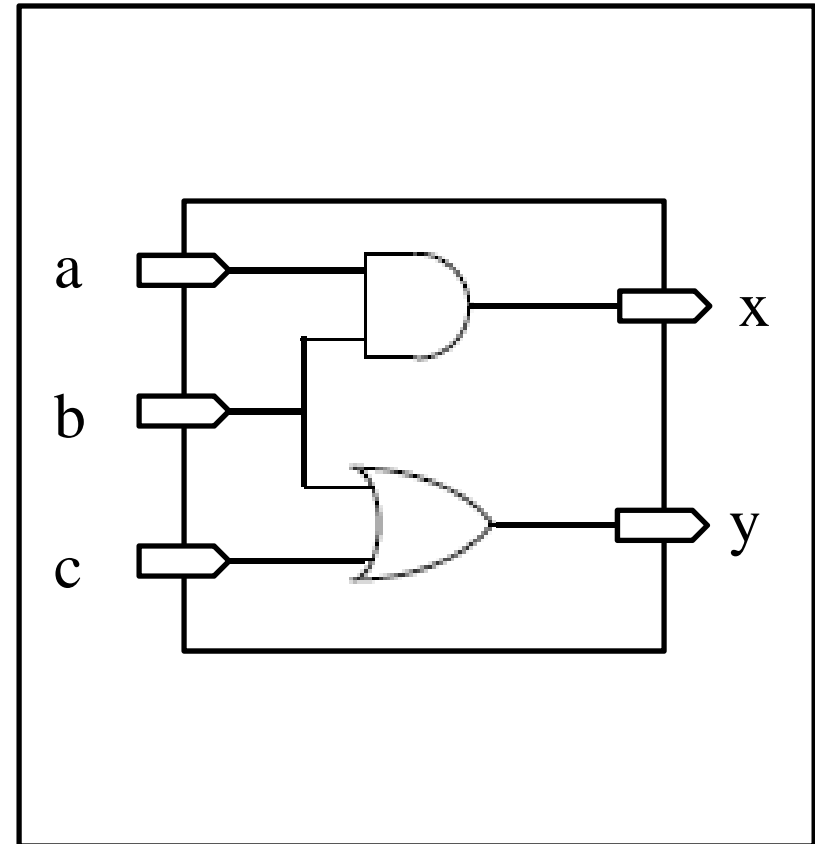
```
BEGIN
```

```
  x <= a AND b;
```

```
  y <= c OR b;
```

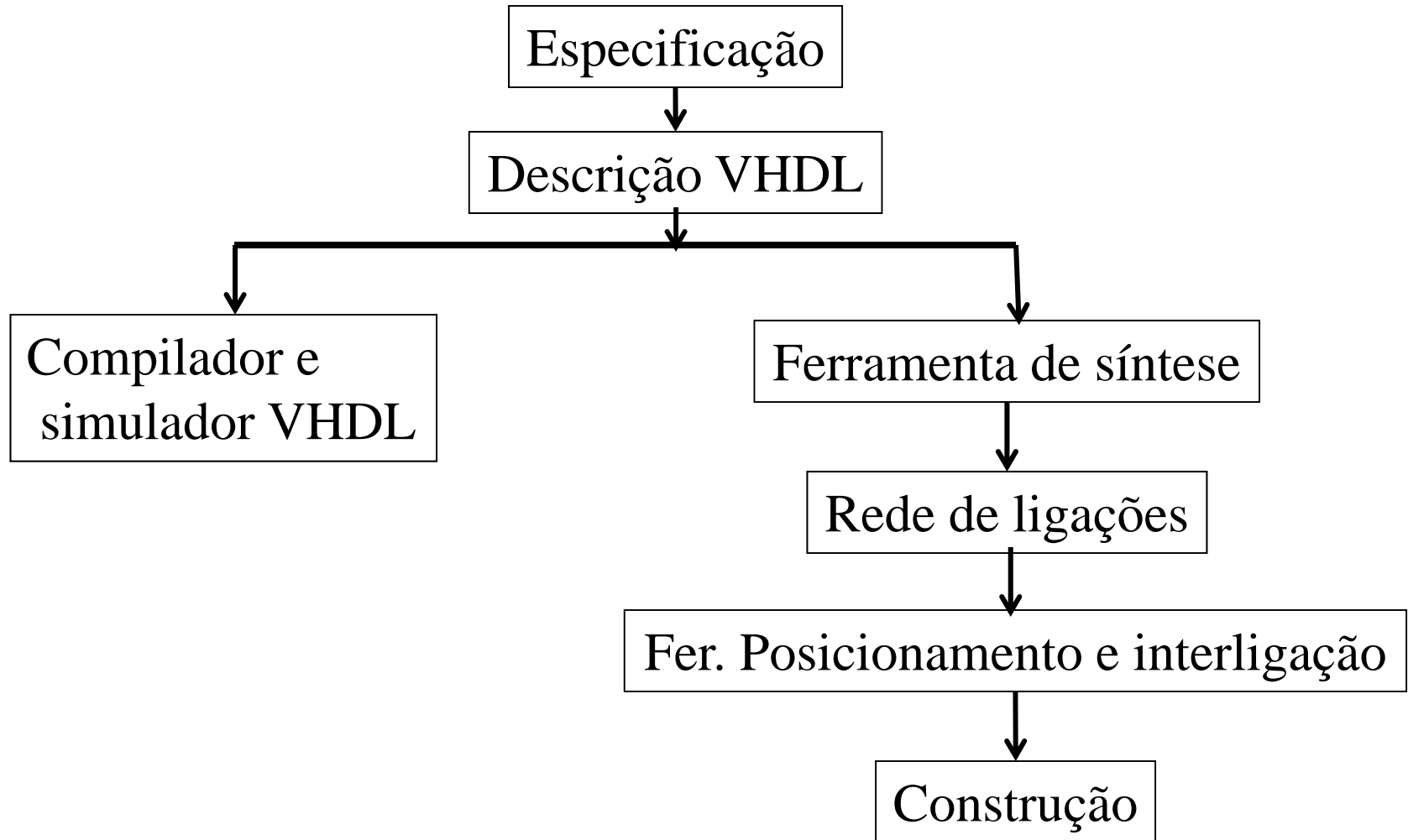
```
END teste
```

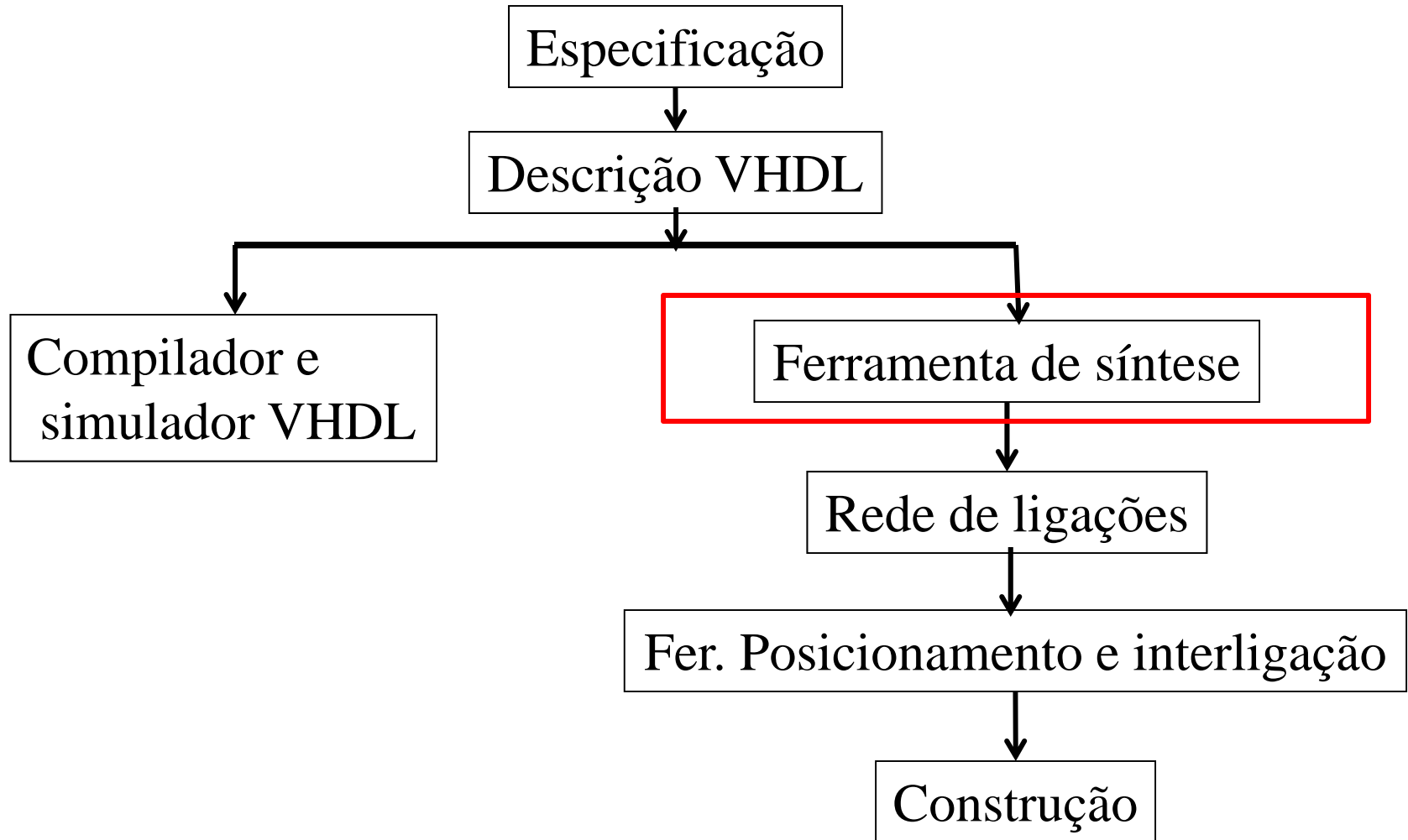
Descrição VHDL

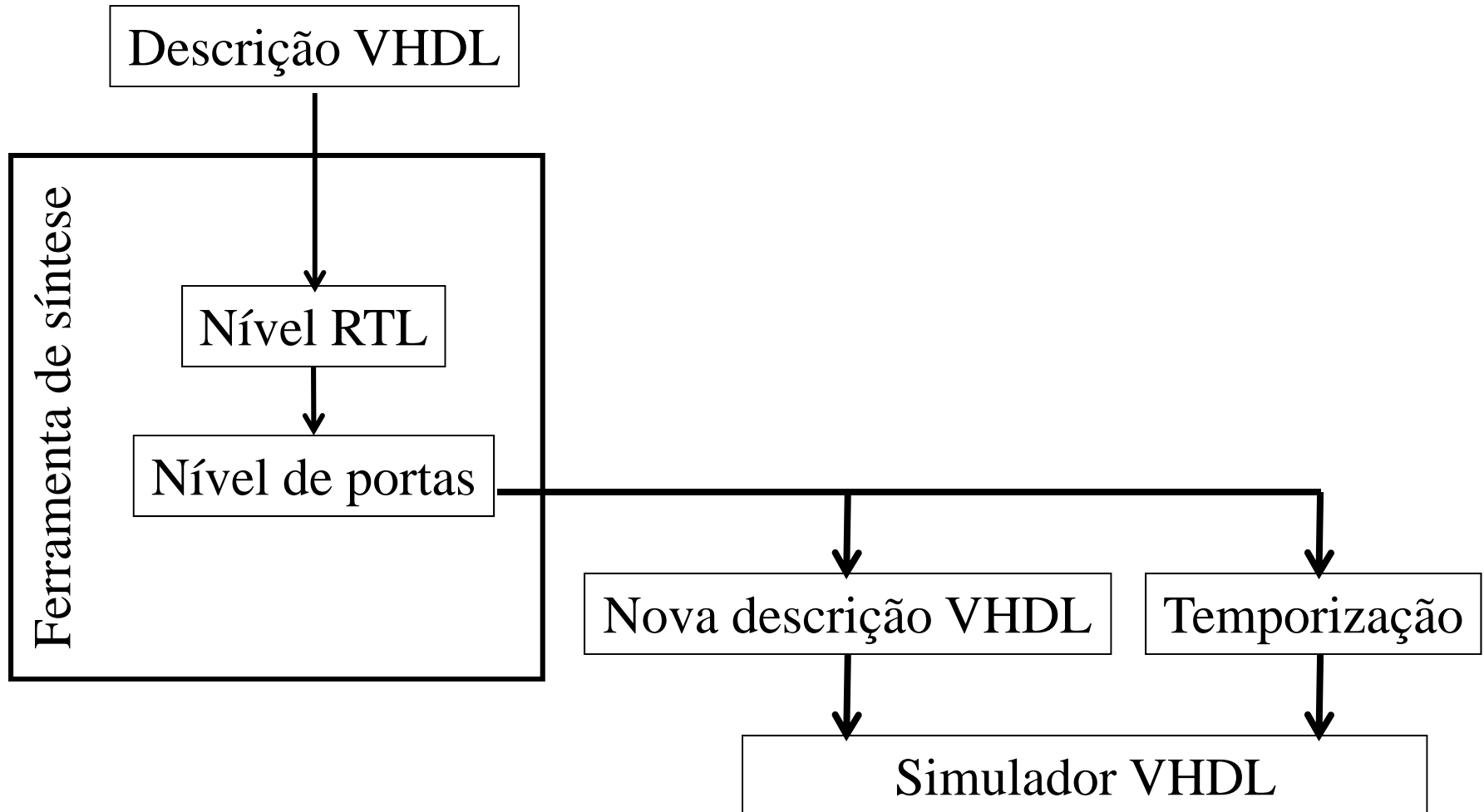


Circuito digital

Etapas de projeto VHDL







- *Register Transfer Level* (RTL)
 - Sistema digital é visto como um subsistema de dados e um subsistema de controle.
 - O estado do subsistema de dados é definido pelo conteúdo dos registradores.
 - Função do sistema é realizada por uma seqüência de transferências de registradores.
 - Essa seqüência é controlada pelo subsistema de controle.
 - Independente da tecnologia alvo.
 - Representação em forma de grafos de execução.

- Avaliação do polinômio:

$$P_7 = \sum_{i=0}^7 p_i x^i$$

- Forma 1:

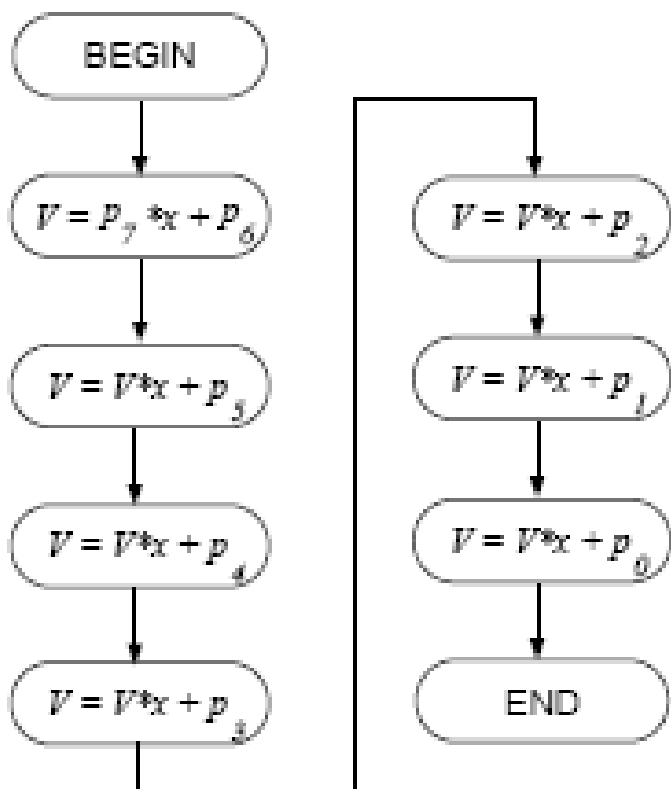
$$P_7 = p_0 + x(p_1 + x(p_2 + x(p_3 + x(p_4 + x(p_5 + x(p_6 + xp_7))))))$$

- Forma 2:

$$P_7 = (p_0 + xp_1) + (p_2 + xp_3)x^2 + [(p_4 + xp_5) + x^2(p_6 + xp_7)](x^2)(x^2)$$

Síntese de alto nível - EXEMPLO

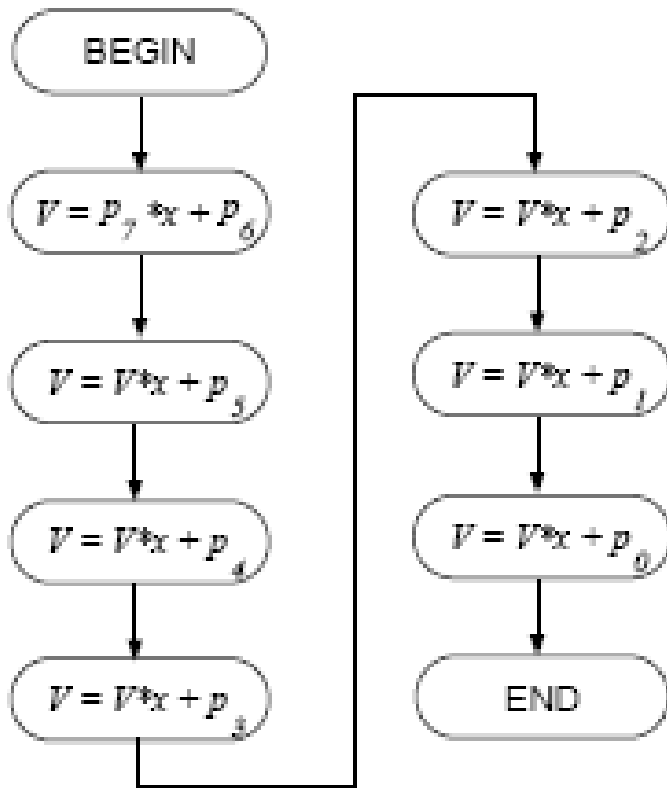
$$P_7 = p_0 + x(p_1 + x(p_2 + x(p_3 + x(p_4 + x(p_5 + x(p_6 + xp_7))))))$$



...($p_5 + x($ V $)$

Síntese de alto nível - EXEMPLO

$$P_7 = p_0 + x(p_1 + x(p_2 + x(p_3 + x(p_4 + x(p_5 + x(p_6 + xp_7))))))$$



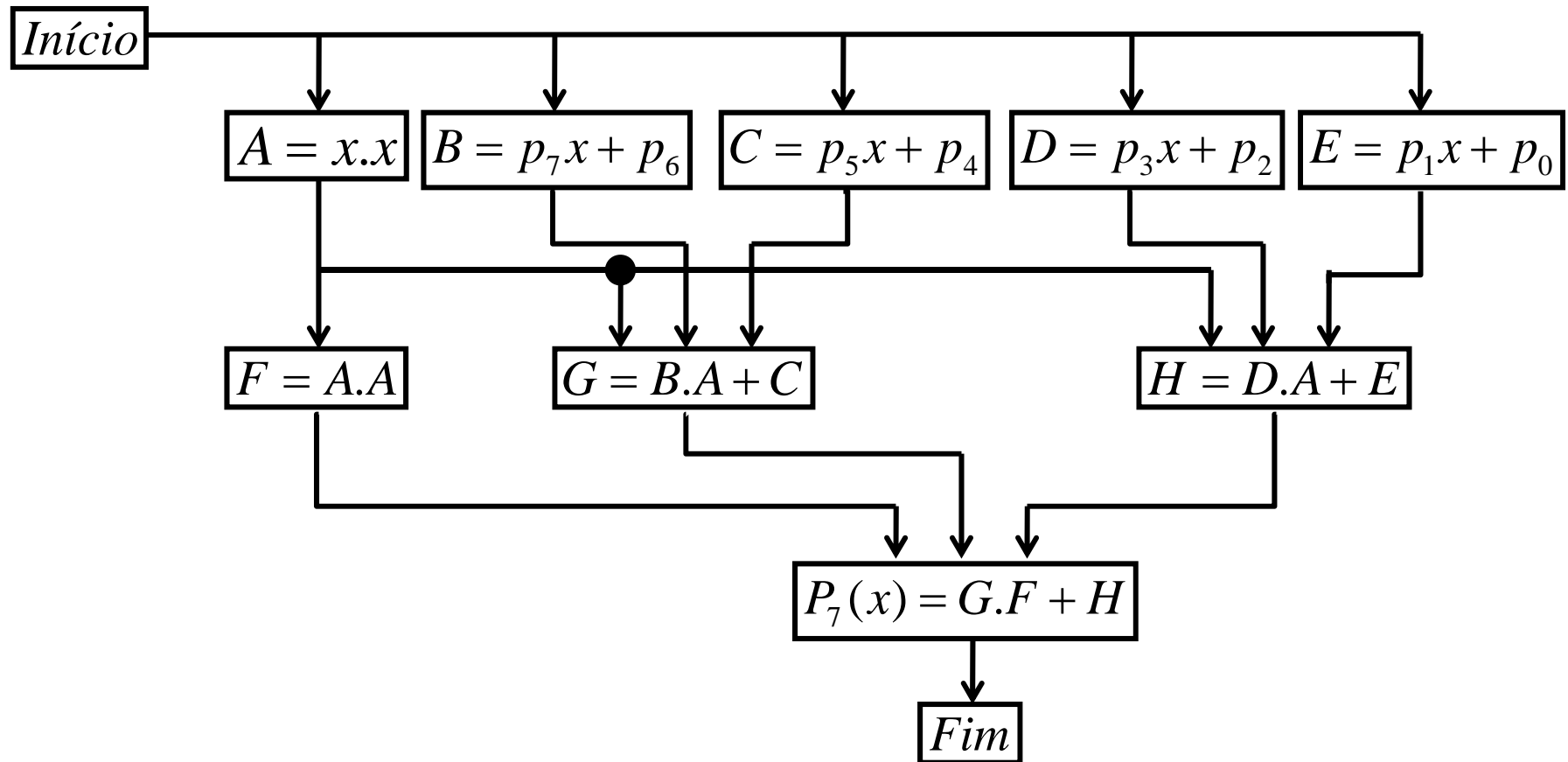
$$...(p_5 + x(V))...$$

Grafo seqüencial:

- Pouco hardware (1 elemento)
- Tempo de execução elevado (7 passos)

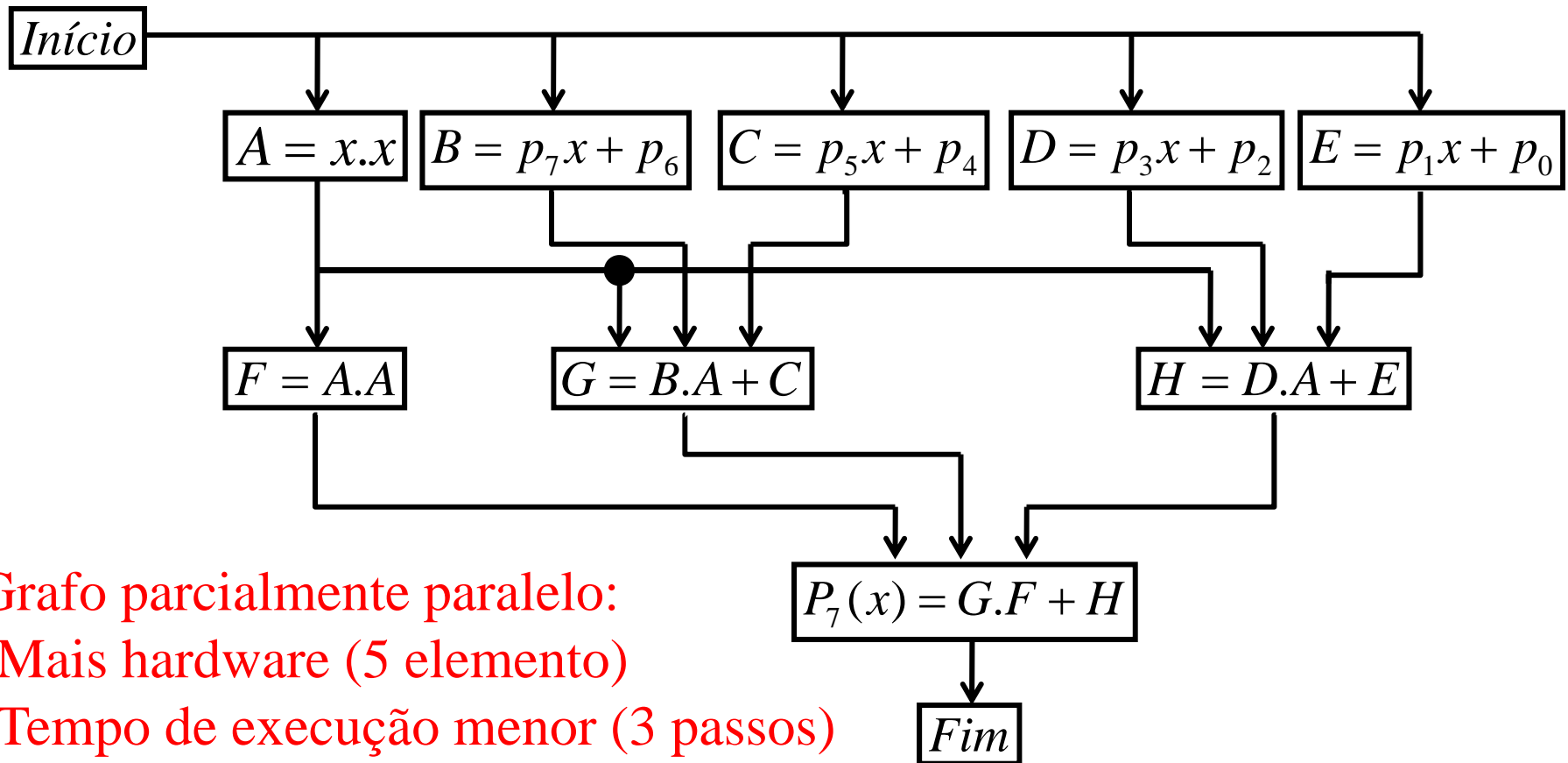
Síntese de alto nível - EXEMPLO

$$P_7 = (p_0 + xp_1) + (p_2 + xp_3)x^2 + [(p_4 + xp_5) + x^2(p_6 + xp_7)](x^2)(x^2)$$



Síntese de alto nível - EXEMPLO

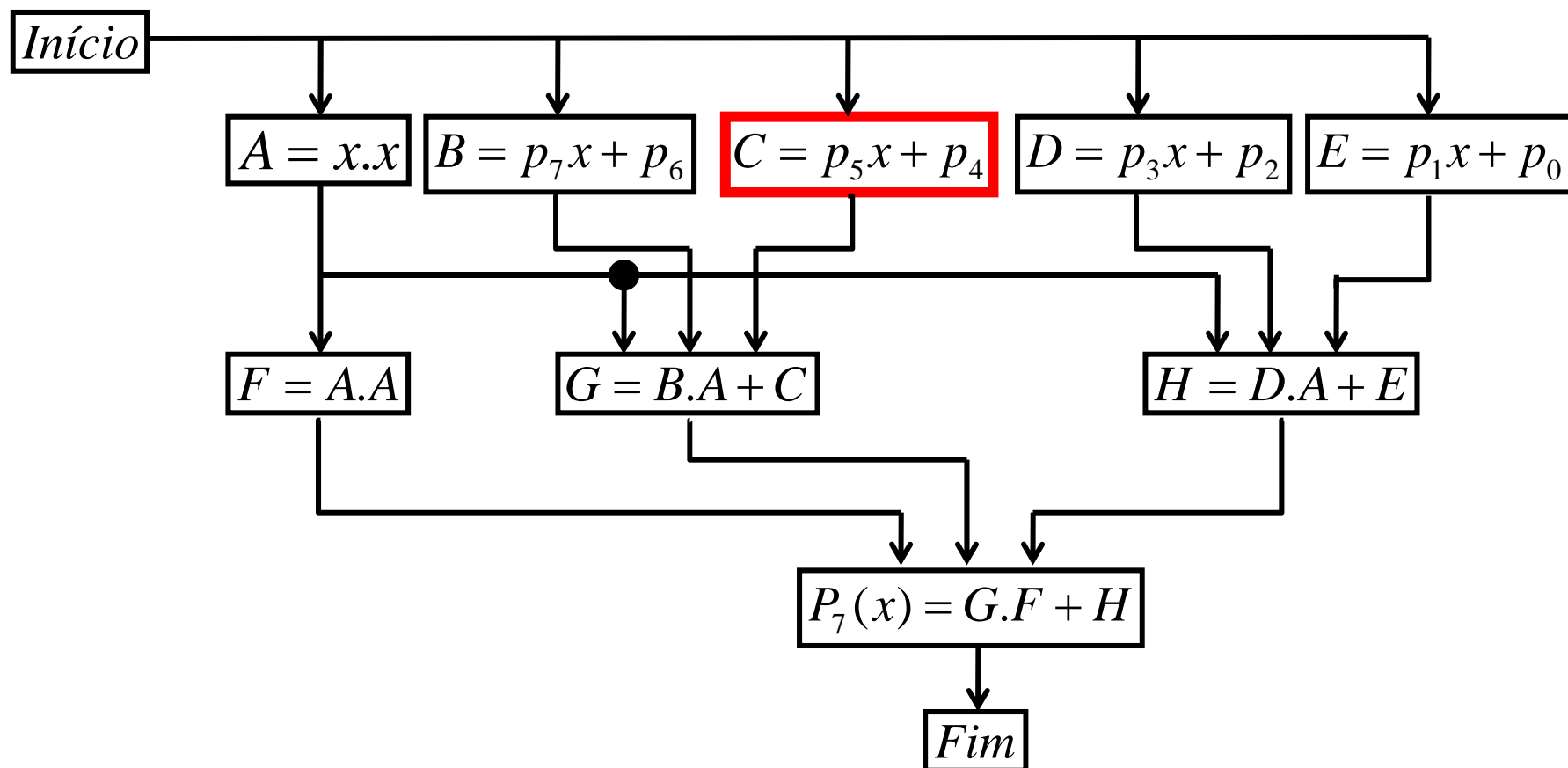
$$P_7 = (p_0 + xp_1) + (p_2 + xp_3)x^2 + [(p_4 + xp_5) + x^2(p_6 + xp_7)](x^2)(x^2)$$

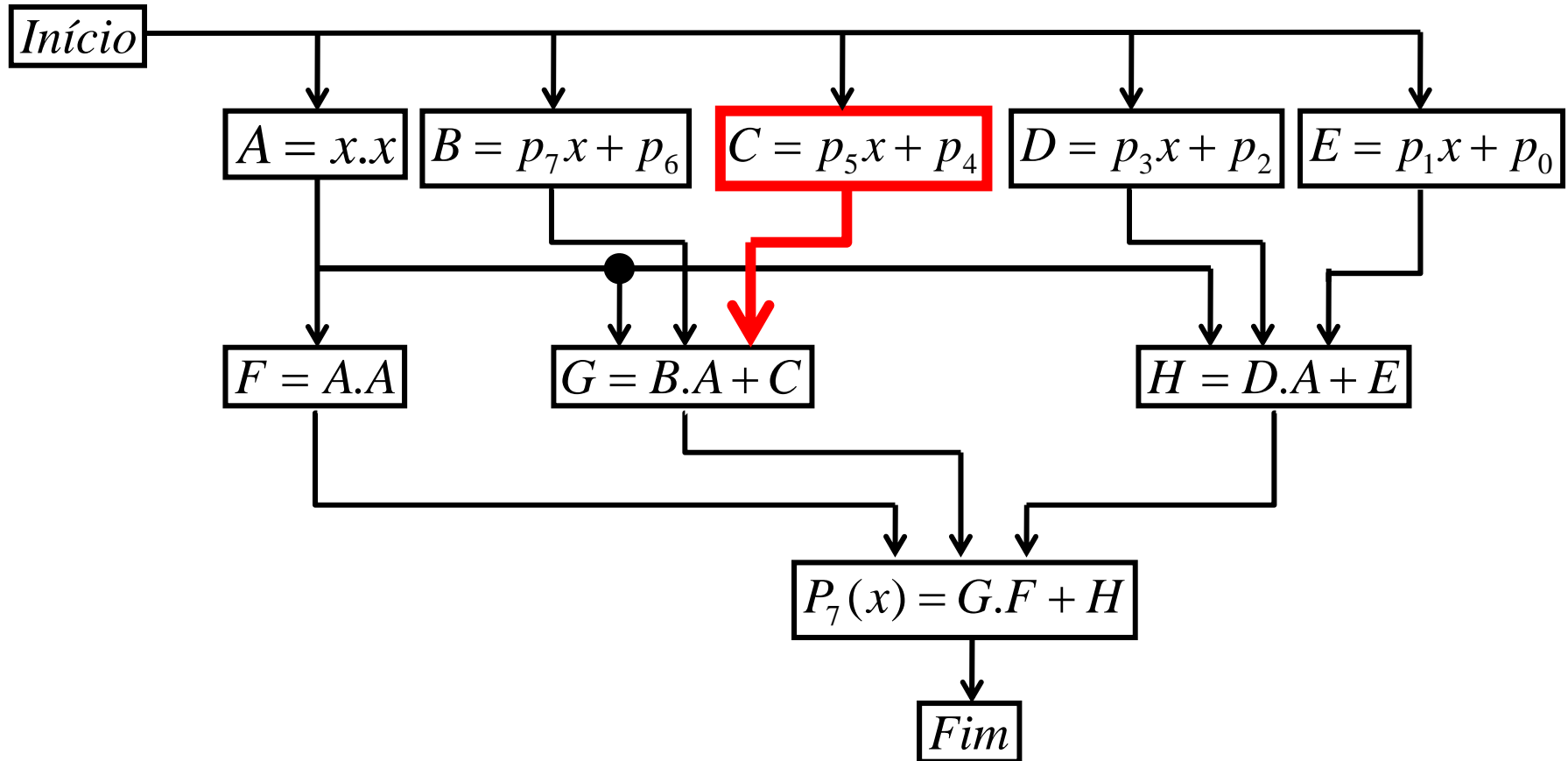


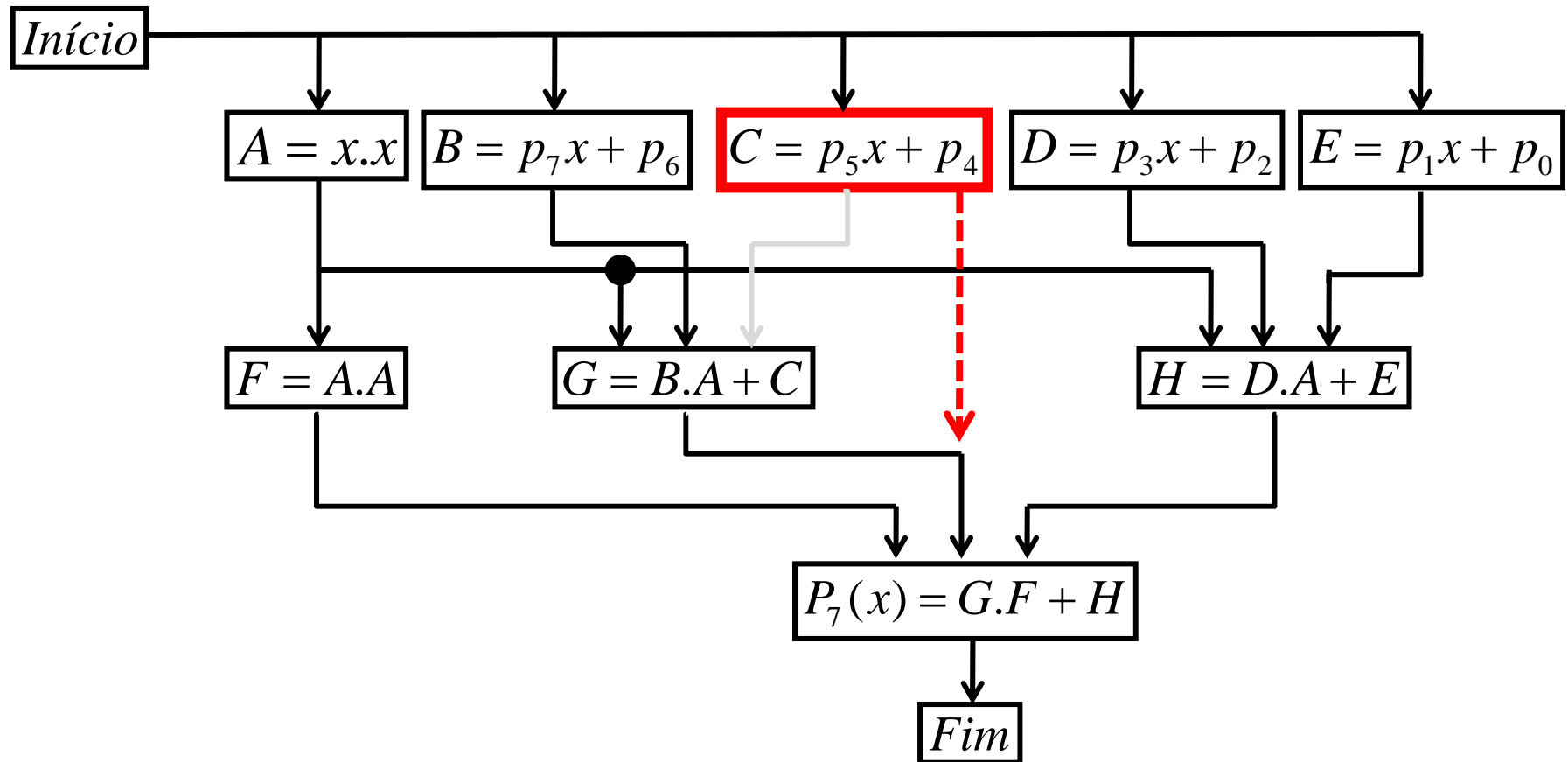
Grafo parcialmente paralelo:

- Mais hardware (5 elemento)
- Tempo de execução menor (3 passos)

- *Scheduling* (Agendamento)
 - Objetivo: Redução do número de passos de controle.
- Alocação
 - Objetivo: Redução do hardware necessário.
 - Unidades funcionais.
 - Memória.
 - Caminhos de comunicação.







- Entradas de códigos em VHDL, Verilog e esquemático.
- Análise de potência e tempo de atrasos.
- Visualizador de RTL.
- Compilador incremental para otimização
- Síntese física.
- Exploração de espaço de projeto.
- Simulação.

Ambiente de desenvolvimento e teste

