



# MC 602

IC/Unicamp  
2011s2  
Prof Mario Côrtes

# VHDL

## Introdução

# Tópicos

- Estruturas básicas do VHDL
- Modelagem de um circuito simples
- Simulação



# VHDL: introdução

- Linguagem de descrição de hardware: suporte para simulação e síntese (padrão IEEE)
- Como representar circuito combinacional simples?

# Modelo completo de um circuito

```
Library IEEE;  
use IEEE.std_logic_1164.all;
```

```
Entity exemplo IS
```

```
    Port (a, b, c : IN std_logic;  
          f : OUT std_logic);
```

```
End exemplo;
```

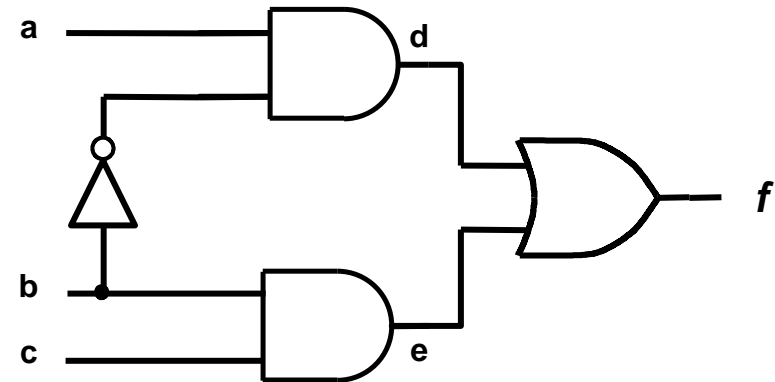
```
Architecture estrutural OF exemplo IS
```

```
    signal d, e : std_logic;
```

```
Begin
```

```
    f <= d or e;  
    d <= a and not(b);  
    e <= b and c;
```

```
End estrutural
```





# Principais blocos

```
Library IEEE;  
use IEEE.std_logic_1164.all;
```

Cabeçalho:

- bibliotecas em uso

Entity exemplo IS

```
Port (a, b, c : IN std_logic;  
      f : OUT std_logic);  
End exemplo;
```

Entity:

- Define o nome
- Define as interfaces
- Ports Inputs/Outputs
- Tipos de sinal

Architecture estrutural OF exemplo IS

```
signal d, e : std_logic;  
Begin  
  f <= d or e;  
  d <= a and (not b);  
  e <= b and c;  
End estrutural
```

Architecture:

- Descreve conteúdo funcional do componente
- Possíveis mais de uma
- Definição de sinais internos
- Atribuição de sinais
- Ordem importa??

# Conceitos básicos

- Sinais (no exemplo são os sinais: a, b, c, d, e, f)
  - Representam os “fios” do circuito
- Alguns tipos dos sinais

```
type bit is ('0', '1');
```

```
type std_logic is (
```

```
    'U', -- não iniciado (unitialized)
```

```
    'X', -- desconhecido (unknow) forte
```

```
    '0', -- zero forte
```

```
    '1', -- um forte
```

```
    'Z', -- alta impedância (tri-state)
```

```
    'W', -- desconhecido fraco
```

```
    'L', -- zero fraco
```

```
    'H', -- um fraco
```

```
    '-' ); -- indiferente (don't care)
```

-- inicia um comentário



# Tutorial de VHDL da Altera

- Resumo do tutorial disponível no link tutoriais do site do curso
  - `tut_quartus_intro_vhdl.pdf`

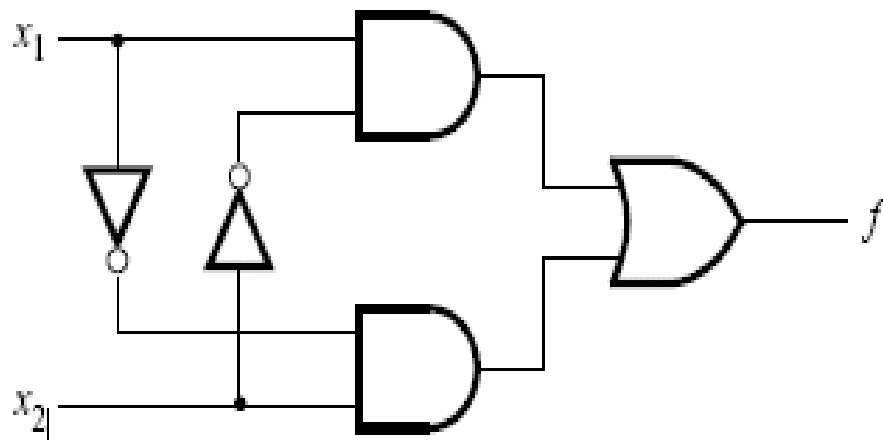


## Criação de novo projeto (ver diff)

- Abrir o Quartus
- File > New Project Wizard
  - Definir diretório onde o projeto será armazenado:  
`tut_vhdl`
  - Escolher nome do projeto (2 próximos campos):  
`light_vhdl`
  - Next
  - Next (mecanismo para adicionar arquivos)
  - Family device settings:
    - escolher Cyclone II EP2C20F484C7
  - Next (other EDA tools)
  - Finish



# Circuito a ser criado



$x_1$	$x_2$	$f$
0	0	0
0	1	1
1	0	1
1	1	0

Figure 11. The light controller circuit.

# Criação do código VHDL

- File > New > **VHDL File**
- File > Save as > **light\_vhdl** > OK
  - Atenção: marcar checkbox “Add file to current project”
- Copiar e colar texto abaixo e salvar

```
LIBRARY ieee ;  
USE ieee.std_logic_1164.all ;
```

```
ENTITY light_vhdl IS  
    PORT ( x1, x2 : IN STD_LOGIC ;  
          f : OUT STD_LOGIC ) ;  
END light ;
```

```
ARCHITECTURE LogicFunction OF light_vhdl IS  
BEGIN  
    f <= (x1 AND NOT x2) OR (NOT x1 AND x2);  
END LogicFunction ;
```

# Próximos passos

- são idênticos aos usados para diag. esquemático
  - compilação
  - atribuição de pinos
  - criação de waveform
  - simulação
  - programação



# Compilação e atribuição de pinos

- Compilação
  - Processing > Start Compilation, ou clicar no ícone ►
  - Verificar o relatório de compilação: Processing > Compilation Report, ou clicar no ícone apropriado
- Atribuição de pinos
  - (associar um pino da FPGA a um dispositivo de entrada/saída da placa)
  - (ver tabela de atribuição de pinos no manual de usuário ou no arquivo DE1pin assignments.odt)
  - Entradas: SW0 e SW1, associadas aos pinos PIN\_L22 e PIN\_L21
  - Saídas: LED verde DG0, associado ao pino PIN\_U22
  - Assignments > Pins: selecionar os pinos acima e OK

# Criação de waveform

- Recompilar
- Criação de waveform para simulação (ver tutorial):
  - File > New > Vector Waveform File
  - Clicar botão direito na coluna de nomes de sinais > Insert > Insert Node > Node Finder
  - List > selecionar os sinais e movê-los para o painel direito
  - Edit > End Time > 200 ns
  - Inserir forma periódica para as entradas
    - Selecionar linha x1; pressionar botão “Count Value” > iniciar 1, a cada 100ns
    - Repetir para x2, com iniciar 0, a cada 40ns
  - Salvar

# Simulação

- Simulação funcional: atrasos não são levados em consideração
  - Assignment > Settings > Simulator Settings > Simulation mode = Functional
  - Geração de netlist: Processing > Generate Functional Simulation Netlist
  - Processing > Start Simulation (ou pressionar ícone apropriado)
  - Observar que não há atrasos para a saída
- Simulação com timing: há atrasos
  - Assignment > Settings > Simulator Settings > Simulation mode = Timing
  - Processing > Start Simulation (ou pressionar ícone apropriado)
  - Observar atrasos e glitches (hazards)



# Construções de VHDL vistas nesta aula

- Cabeçalho e bibliotecas
- Entity: significado, ports, tipo de sinais
- Architecture
- Definição de sinais internos (não fazem parte da interface)
- Atribuição de sinais
- Alguns operadores booleanos
- Comandos concorrentes
- Tipos de sinal: bit e std\_logic
- Convenção para comentário ( -- )