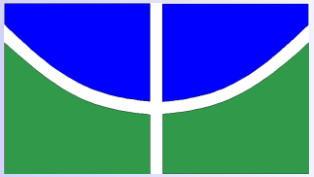


Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciências da Computação

Síntese de Hardware

Verilog

Altera DE1-SoC

Linguagem de Descrição de Hardware (HDL)

- Motivação:

  Sistemas Digitais complexos são praticamente impossíveis de se definir e estudar através do esquemático ao nível de portas lógicas

Verilog

- Juntamente com VHDL é uma das mais populares HDLs
- Histórico

  1984/85 – criada por Philip Moorby (Gateway Design System Co.) adquirida pela Cadence

  1990 – Open Verilog International : Domínio público

  1995 – Padrão IEEE 1364

  2005 – revisão e atualização do padrão, IEEE 1364

  2012 – IEEE 1800, SystemVerilog, orientação a objetos

- Sintaxe semelhante à linguagem C
- Usada para projetos de circuitos ASIC (*Application Specific Integrated Circuit*) e baseados FPGA (*Field Programmable Gate Array*).

Verilog

- Níveis de Abstração
  - Nível de Layout (*Layout Level*)
  - Nível de Portas Lógicas (*Gate Level*)
  - Nível de Transferência de Registros (*Register Transfer Level*)
  - Nível Comportamental (*Behaviour Level*)

Etapas de um projeto

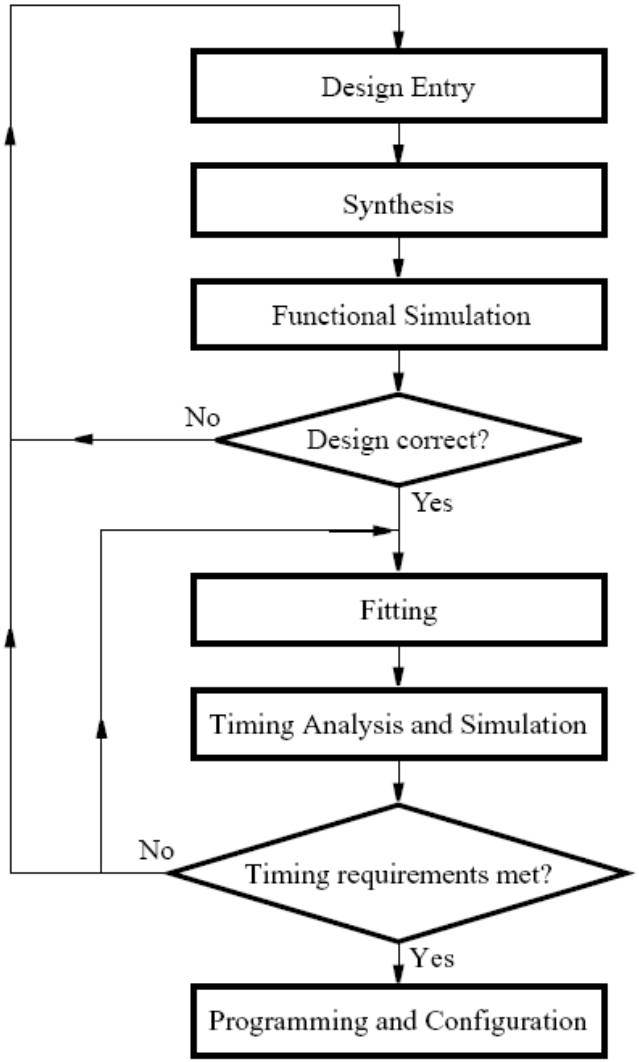
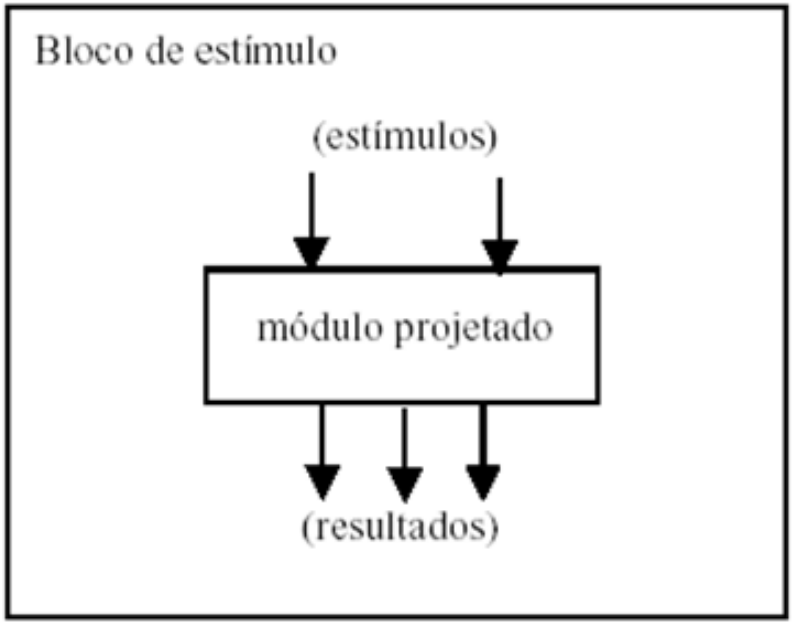


Figure 1. Typical CAD flow.

Estilos de Simulação e Teste (*testbench*)



Verilog – aspectos básicos

- Representação de Números:

  <tamanho>’<base><número>

  Ex.: 1’b1 8’hFF 16’d65535

- Valores: [0,1]: níveis lógicos

  x : desconhecido, z: alta impedância

  Ex.: 32’bz 8’h0x 4’b1z0x

- Tipos: wire, reg, como input, output, inout

  Ex.: input [31:0] A; output wire x; reg [3:0] vetor[0:15]

- Tipos abstratos: integer, real (IEEE 754), time

Operações

arithmetic operator

operator	operation
+	arithmetic addition
-	arithmetic substraction
*	arithmetic multiplication
/	arithmetic division
%	arithmetic modulus

other operators

operator	operation
>>	logical shift right
<<	logical shift left
==, !=	equality
===, !==	identity
?:	conditional
{}	concatenate
{{}}	replicate

unary operator (1-bit result)

operator	operation
&	unary reduction AND
~&	unary reduction NAND
|	unary reduction OR
~|	unary reduction NOR
^	unary reduction XOR
~^	unary reduction XNOR

– unary operation will preform the operation on each bit of the operand and get a one-bit result.

|8'b00101101 is 1'b1

bit-wise operators

operator	operation
~	bit-wise NOT
&	bit-wise AND
|	bit-wise OR
^	bit-wise XOR
~^	bit-wise XNOR

– binary bit-wise operation will perform the operation one bit of a operand and its equivalent bit on the other operand to calculate one bit for the result.

(8'b11110000 & 8'b00101101) is 8'b00100000

logical operators

operator	operation
!	logical NOT
&&	logical AND
||	logical OR
==	logical equality
!=	logical inequality
===	logical identity
!==	the inverse of ===

– logical operator operate with logic values. (non-zero is true, and zero value is false).

if(sel == 4'h03) ……. else ……..

FPGA: Intel Cyclone-V 5CSEMA5F31C6

Possui:

- Chip com 896 pinos
- 32.070 Adaptive Logic Module (ALM)
  - Função Lógica de 8 entradas
  - 4 registradores de 1 bit
  - Somadores encadeáveis (carry)
  - Roteamento
- Memória RAM 4.065.280 bits
- 87 DSP (multiplicador 18x18, 64 add)
- 6 PLL (*Phase Locked Loop*)
- 288 Pinos de IO para o usuário
- ARM Cortex-A9 dual core

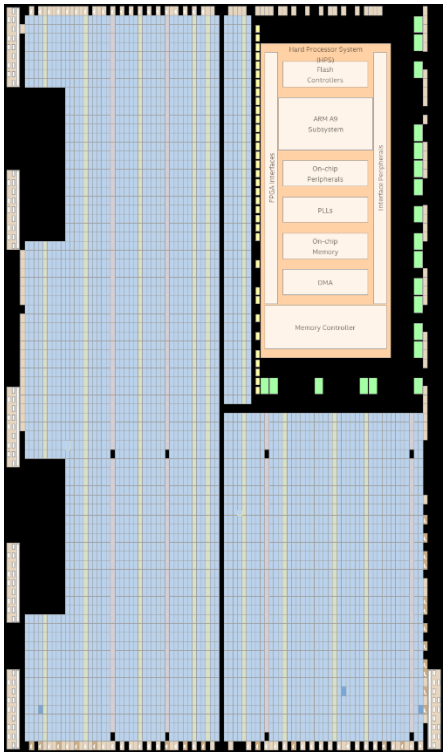
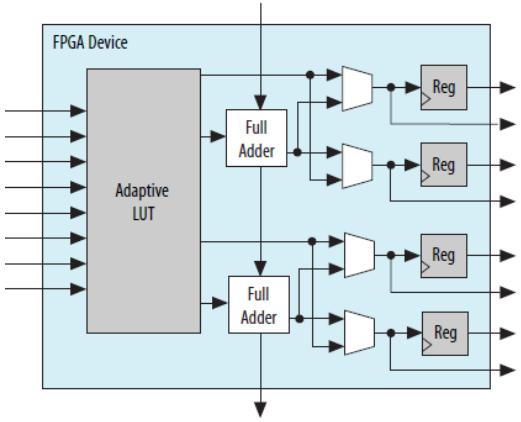
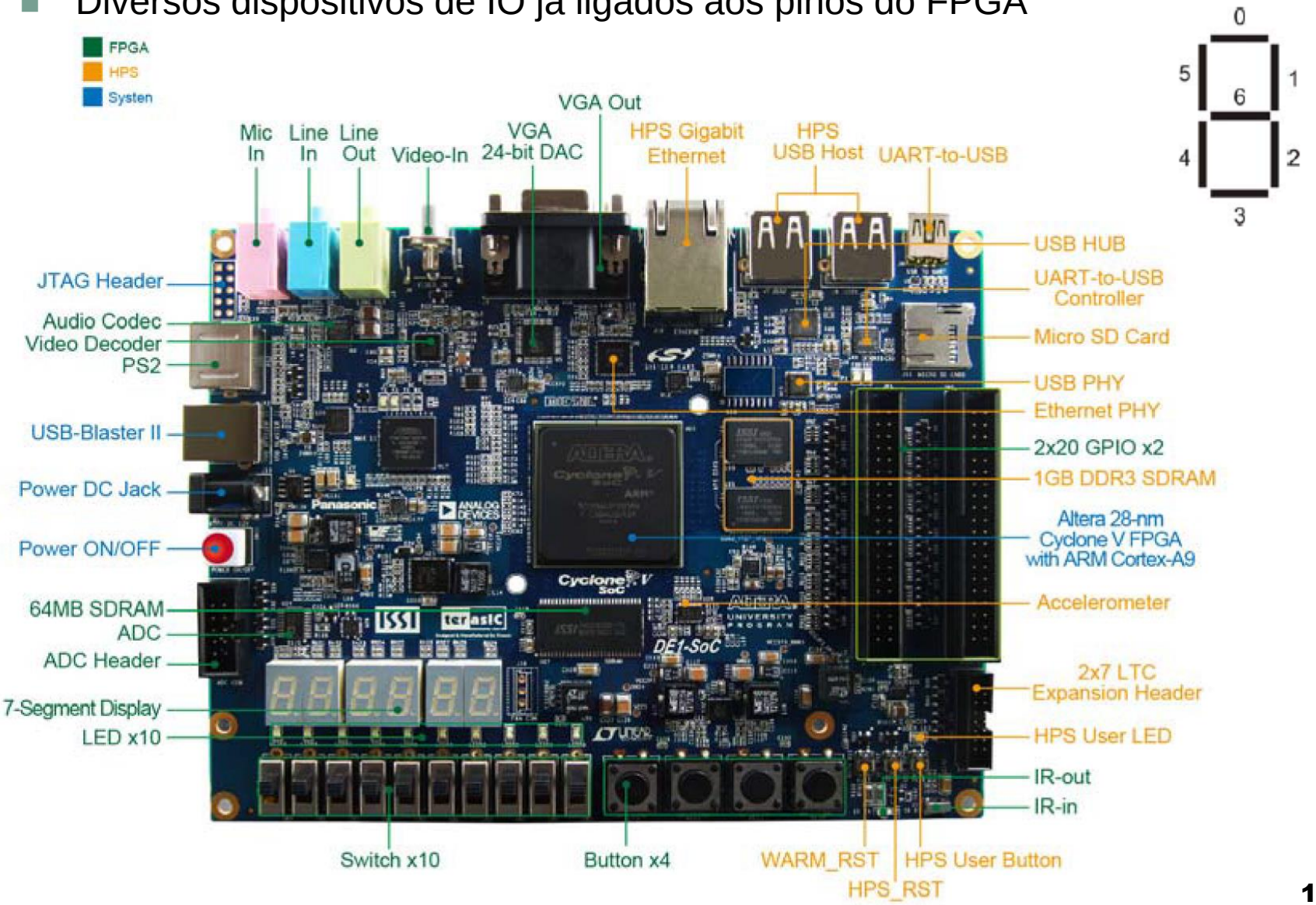


Figure 8: ALM for Cyclone V Devices

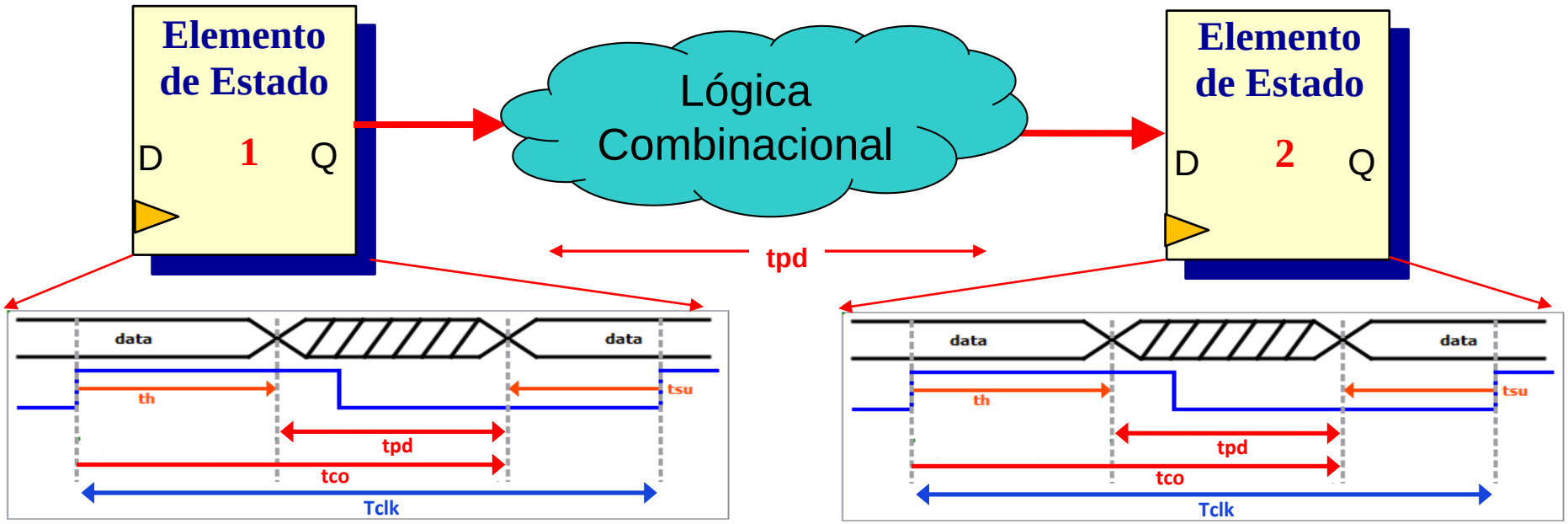


Plataforma Intel DE1-SoC

- Diversos dispositivos de IO já ligados aos pinos do FPGA

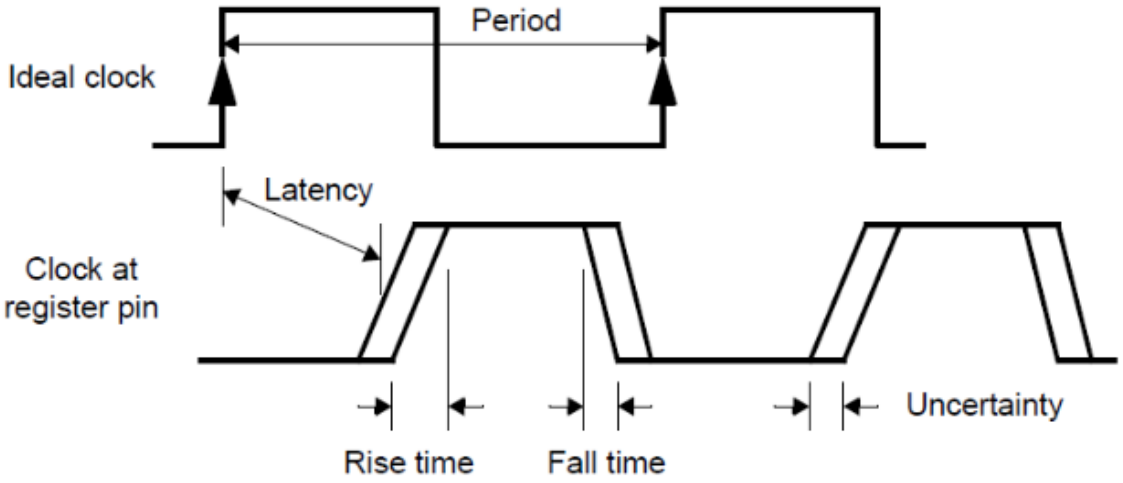


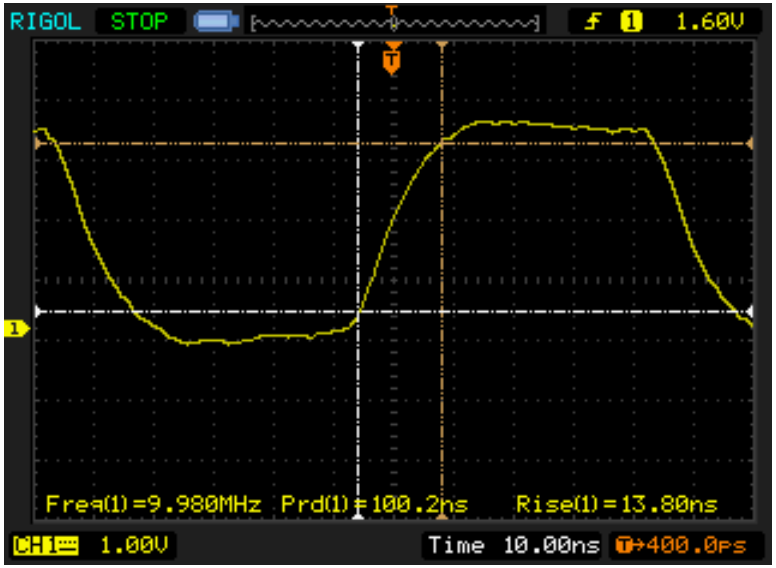
Revisão: Temporização em circuitos digitais



- **A escrita no elemento de estado 1 deve respeitar os tempos de:**
  - tempo de pré-carga (*setup time,* tsu) do elemento 1
  - tempo de hold (*hold time, th*) do elemento 1
- **O elemento de estado 2 só pode ser escrito depois que os dados em sua entrada estarem estáveis**
  - atraso de propagação da saída dado o clock (*clock to output,* tco) do elemento 1
  - atraso da lógica combinacional (*propagation delay*, tpd=max(tp_{HL},tp_{LH}))
  - tempo de pré-carga (*setup time,* tsu) do elemento de estado 2

Sinal de relógio: clock

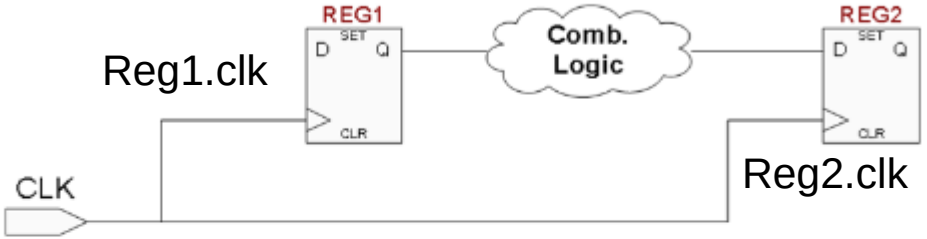


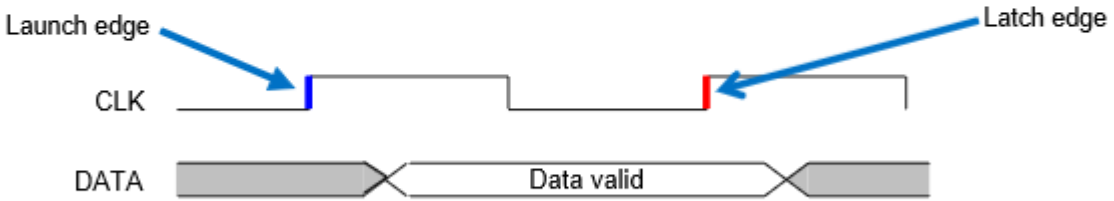


Temporização em circuitos digitais

https://www.altera.com/customertraining/OLT/TQ_Intro_p1/presentation_html5.html

Launch & Latch Edges





Launch edge: the edge which "launches" the data from the source register

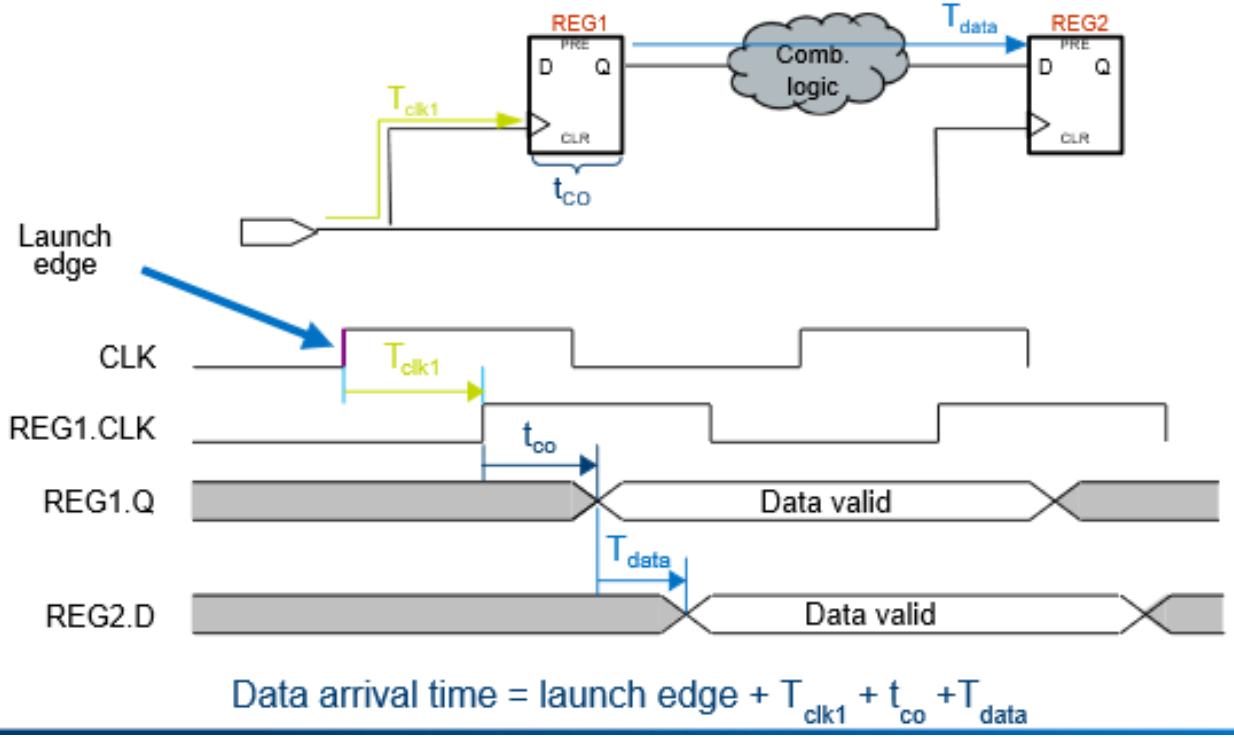
Latch edge: the edge which "latches" the data at the destination register (with respect to the launch edge, selected by timing analyzer; typically 1 clock cycle, i.e. rising to rising edge)

O tempo de chegada do clock nas entradas dos regs geralmente, não são o mesmo! Dependem do caminho! Clock Skew.

Tempo de chegada do dado (tc_{data})

Data Arrival Time

The time for data to arrive at the destination register's (REG2) D input



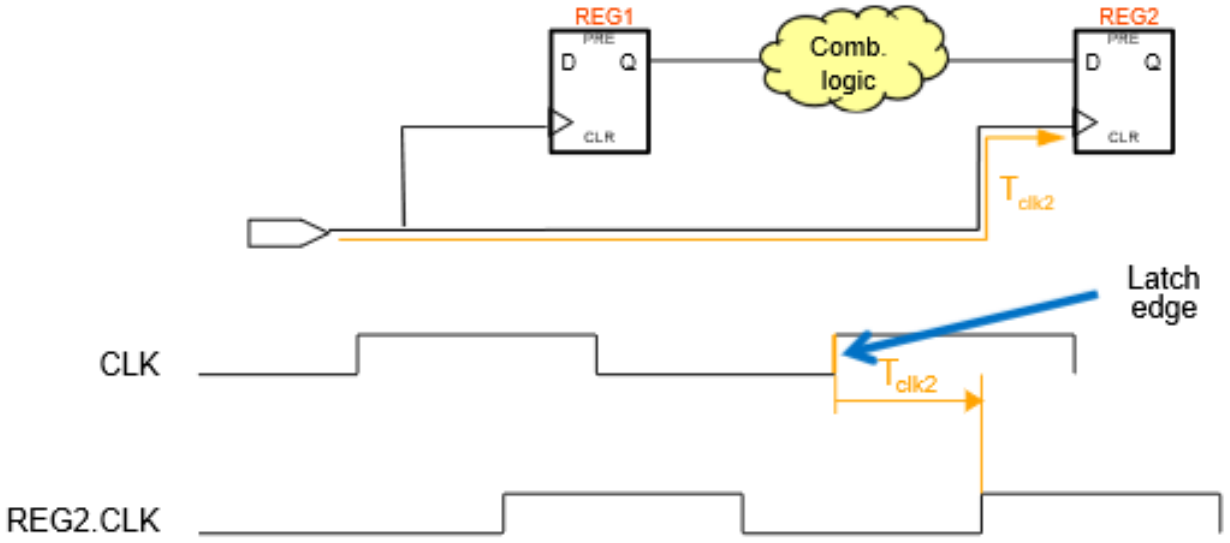
Data arrival time = launch edge + T_{clk1} + t_{co} + T_{data}

$$tc_{data}=t_{clk1}+t_{co}+t_{pd}$$

Tempo de chegada do clock (t_{clock2})

Clock Arrival Time

The time for the clock to arrive at destination register's (REG2) clock input

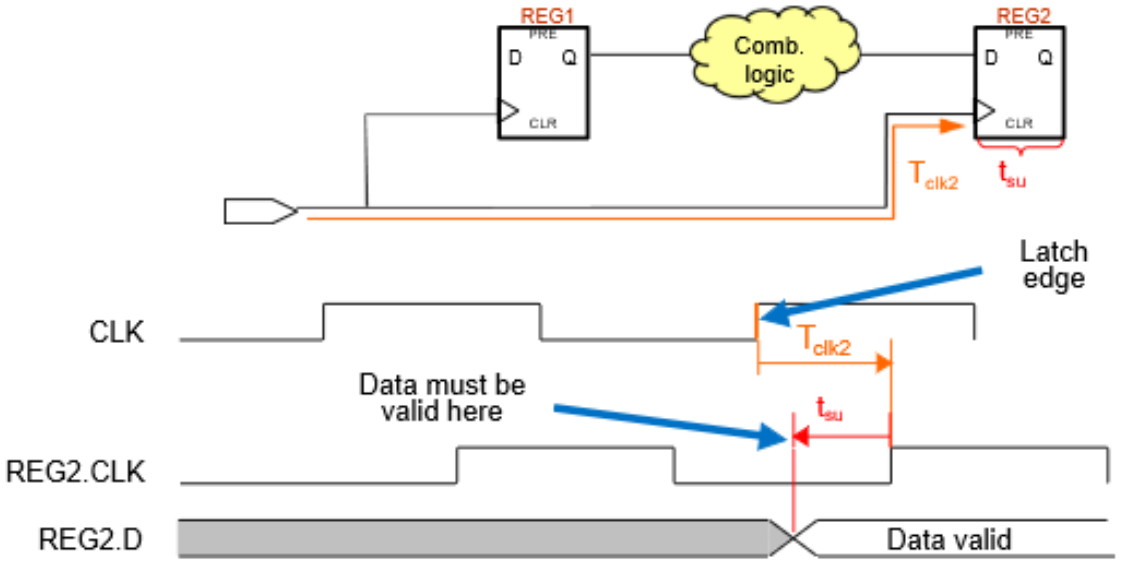


Clock arrival time = latch edge + T_{clk2}

Tempo requerido de setup (tr_{setup})

Data Required Time (Setup)

The minimum time required for the data to be valid before the latch edge so the data can be successfully latched into the destination register (REG2)



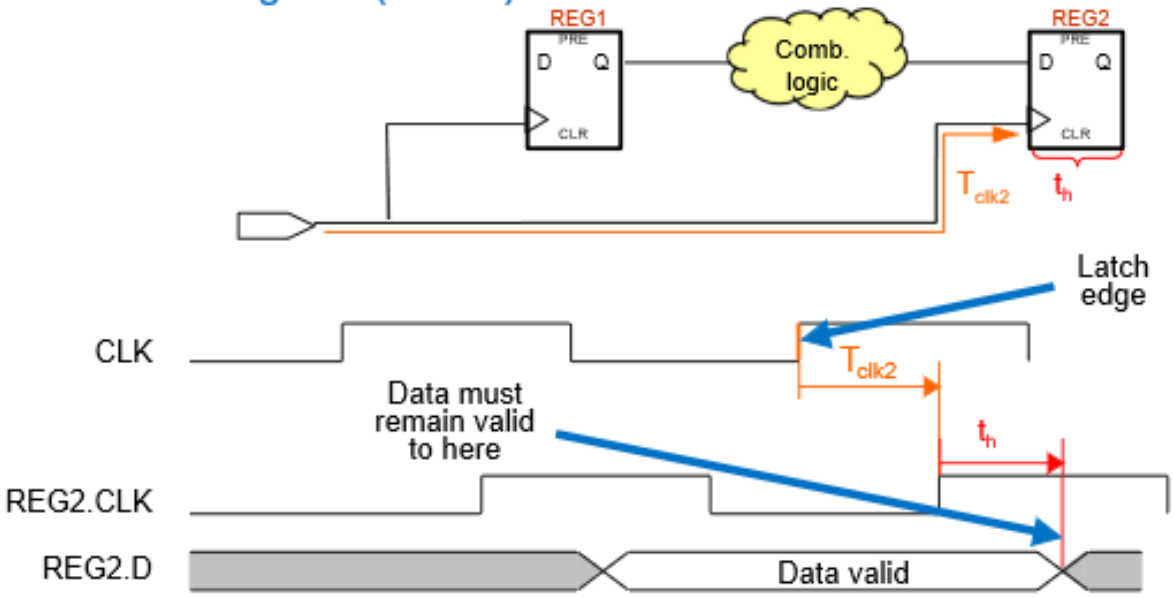
Data required time (Setup) = Clock arrival time - t_{su} - Setup uncertainty (clock Jitter)

$$tr_{setup} = t_{clk2} - t_{su} - \text{incerteza}$$

Tempo requerido de hold (tr_{hold})

Data Required Time (Hold)

The minimum time required after the latch edge for the data to remain valid for successful latching into the destination register (REG2)

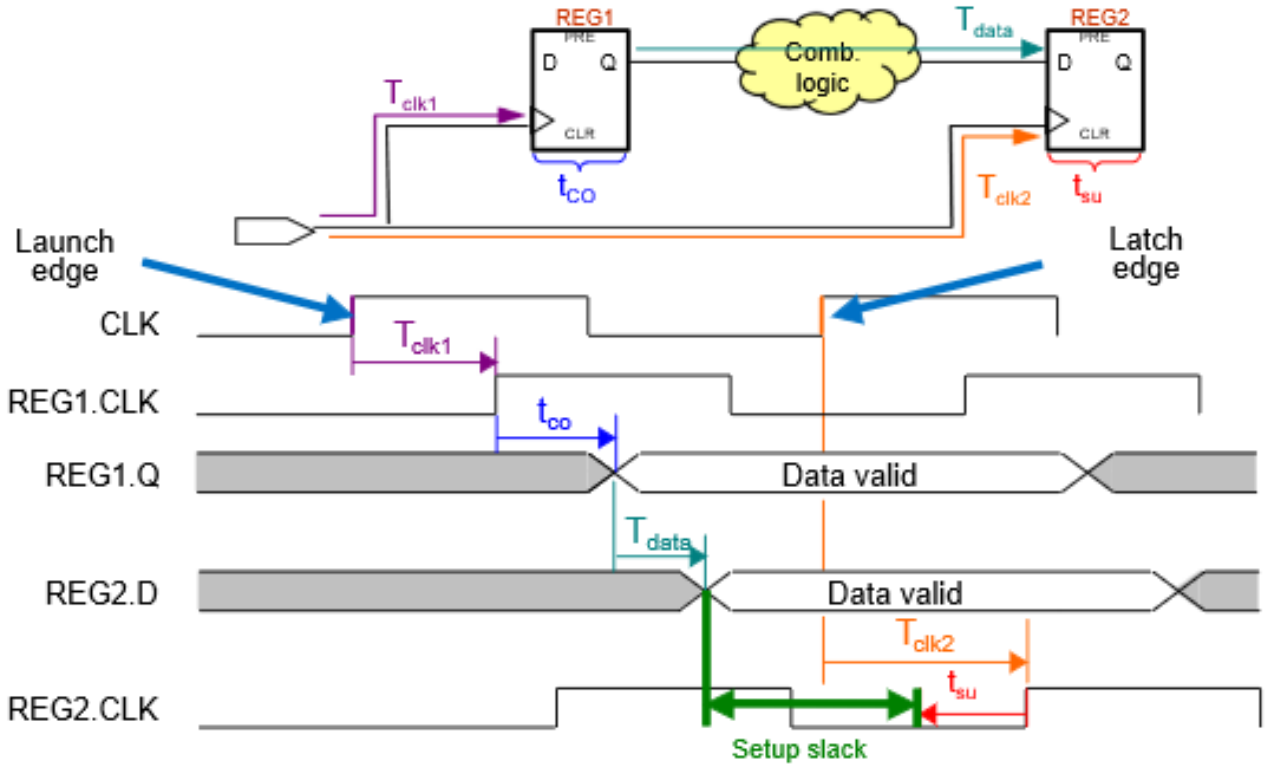


Data required time (Hold) = Clock arrival time + t_h + Hold uncertainty (clock jitter)

$$tr_{hold} = t_{clk2} + t_h + \text{incerteza}$$

Tempo de folga de setup (S_{setup})

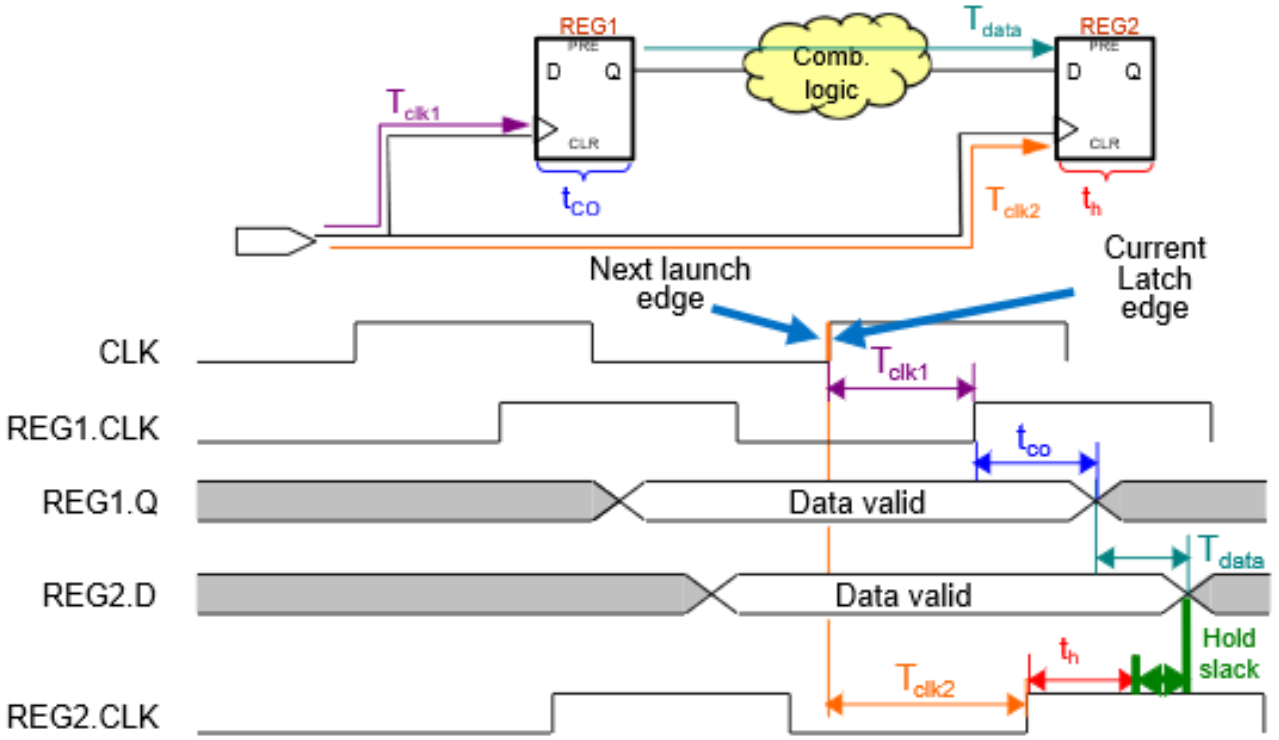
Setup Slack (2)



$$S_{setup} = Min\{tr_{setup}\} - Max\{tc_{data}\}$$

Tempo de folga de hold (S_{hold})

Hold Slack (2)



$$S_{hold} = Min\{tc_{data}\} - Max\{tr_{hold}\}$$

Análise dos tempos de folga

- Se os tempos de folga forem positivos → OK!
  - Requerimentos foram atendidos

- Se os tempos de folga forem negativos → Problema!
  - Requerimentos não foram atendidos
  - Precisa rever o projeto e/ou os requerimentos temporais