



## **Aula 12 – Core IP (Memórias, PLL)**



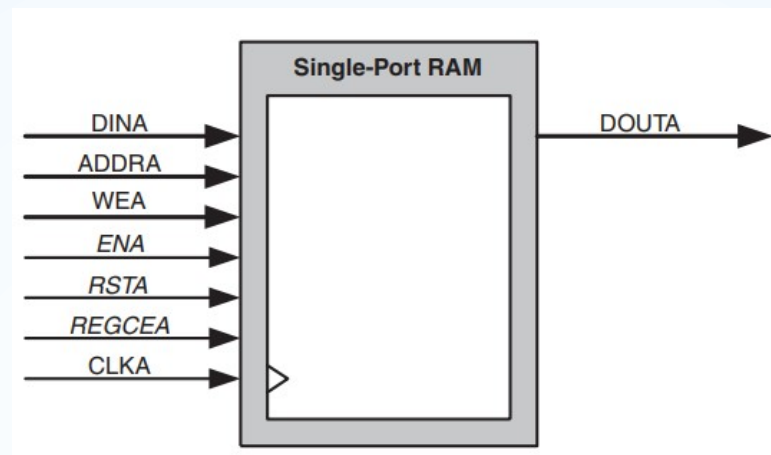
## Tópicos da aula

- **Tipos**
- **Vetores e Matrizes**
- **Memória RAM**
- **Memória ROM**



## Core IP (*Intellectual Property*)

- ❑ São **componentes** (blocos funcionais) desenvolvidos pelos fabricantes de FPGA (Intel, AMD) ou por terceiros
- ❑ São fornecidos na forma de **netlist**
- ❑ Sinais de entrada e saída são documentados
- ❑ Alguns são **gratuitos** outros **pagos** (licença)
- ❑ Uso de ferramentas visuais para configuração (GUI)





Universidade Federal  
de Santa Catarina

## Exemplo de IP: Memória RAM

MegaWizard Plug-In Manager [page 1 of 6]

### RAM: 1-PORT

About Documentation

1 Parameter Settings 2 EDA 3 Summary

Widths/Blk Type/Clocks > Regs/Clock/Byte Enable/Ackrs > Read During Write Option > Mem Init >

Currently selected device family: MAX 10

☒ Match project/default

How wide should the 'q' output bus be? 8 bits

How many 8-bit words of memory? 64 words

Note: You could enter arbitrary values for width and depth

What should the memory block type be?

☒ Auto ☐ MLAB ☐ M9K

☐ M144K ☐ LCs Options...

Set the maximum block depth to Auto words

What clocking method would you like to use?

☒ Single clock ☐ Dual clock: use separate 'input' and 'output' clocks

Resource Usage

1 M9K

Cancel < Back Next > Finish



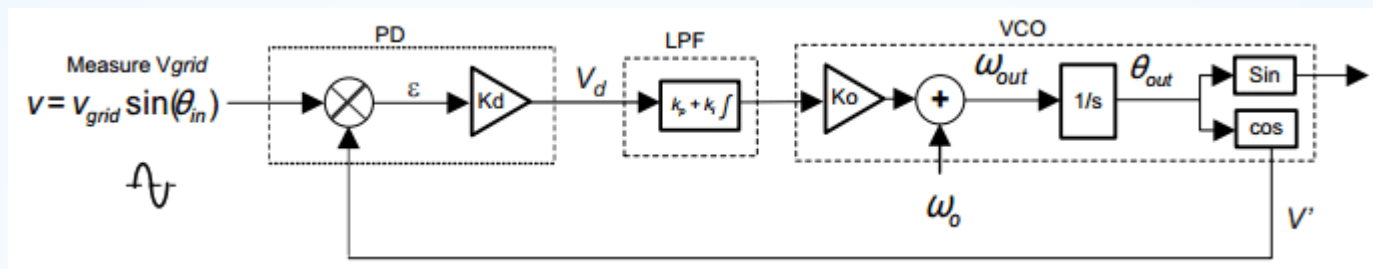
## Exercício

- 1) Implemente uma memória RAM de 16 bits, com 512 words, usando o IP catalog da Intel (Altera).
- 2) Inicialize a memória com um arquivo de dados (.mif ou .hex).



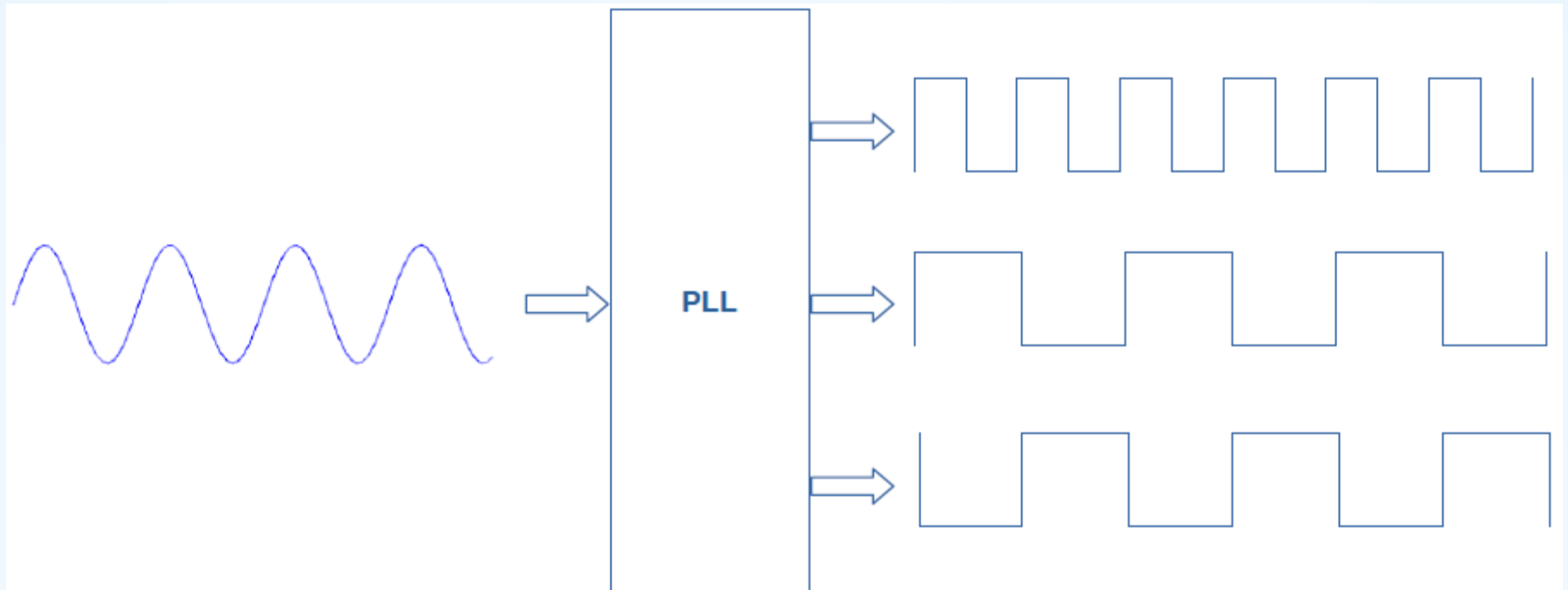
## Controlador PLL

- ❏ **Phase Locked Loop**
- ❏ É um sistema de controle que gera, a partir de um sinal de entrada, um sinal de saída com **frequência** e **fase** instantâneas em **sincronismo** com o sinal amostrado
- ❏ Pode gerar frequências múltiplas a partir do sinal de entrada
- ❏ O diagrama de um sistema PLL básico é composto por, um *Phase Detect* (PD), um *Loop Filter* (LPF) e um *Voltage Controlled Oscillator* (VCO)





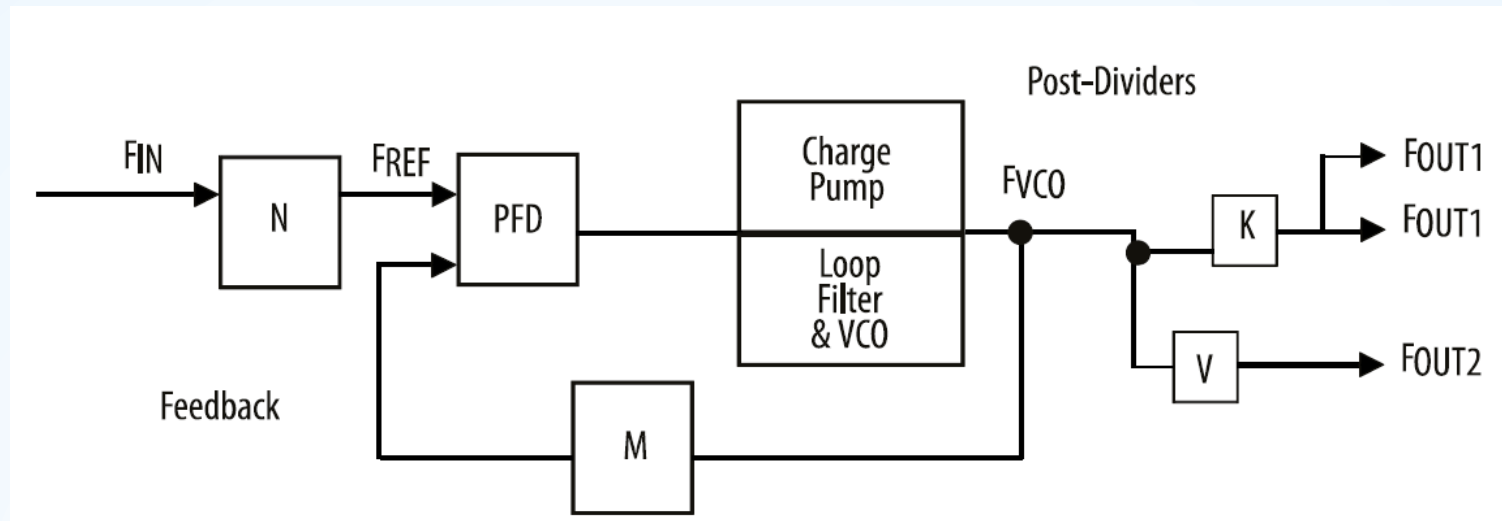
## Exemplo: uso de um PLL





## Core IP: Controlador PLL


- Em geral, FPGAs possuem um ou mais PLLs
- Sinal de entrada normalmente é **senoidal** (uso de cristal ressonador). Sinais gerados pelo PLL são **ondas quadradas**







## Core IP: Controlador PLL

 **ALTPLL**

[About](#) [Documentation](#)

1 Parameter Settings

2 PLL Reconfiguration

3 Output Clocks

4 EDA

5 Summary

clk c0 > clk c1 > clk c2 > clk c3 > clk c4 >

PLL

inclk0  
areset

inclk0 frequency: 100.000 MHz  
Operation Mode: Normal

Clk	Ratio	Ph (dg)	DC (%)
c0	1/1	0.00	50.00

locked

MAX 10

### c0 - Core/External Output Clock

Able to implement the requested PLL

☒ Use this clock

Clock Tap Settings

	Requested Settings	Actual Settings
<input type="radio"/> Enter output clock frequency:	100.00000000 MHz	100.000000
<input checked="" type="radio"/> Enter output clock parameters:		
Clock multiplication factor	1	1
Clock division factor	1	1
Clock phase shift	0.00 deg	0.00
Clock duty cycle (%)	50.00	50.00

Description	Val
Initial VCO phase cycles for M counter	1
VCO phase tap for M counter	0

Note: The displayed internal settings of the PLL is recommended for use by advanced users only

Per Clock Feasibility Indicators

c0 c1 c2 c3 c4

Cancel < Back Next > Finish



## Exercício

Implemente um PLL para gerar os sinal de clock interno e reset (locked) interno, no projeto do Cronômetro.



## **FIM AULA 12**