



- Arquitetura
- RTOS
- Referências

Microcontroladores: (LT36D)

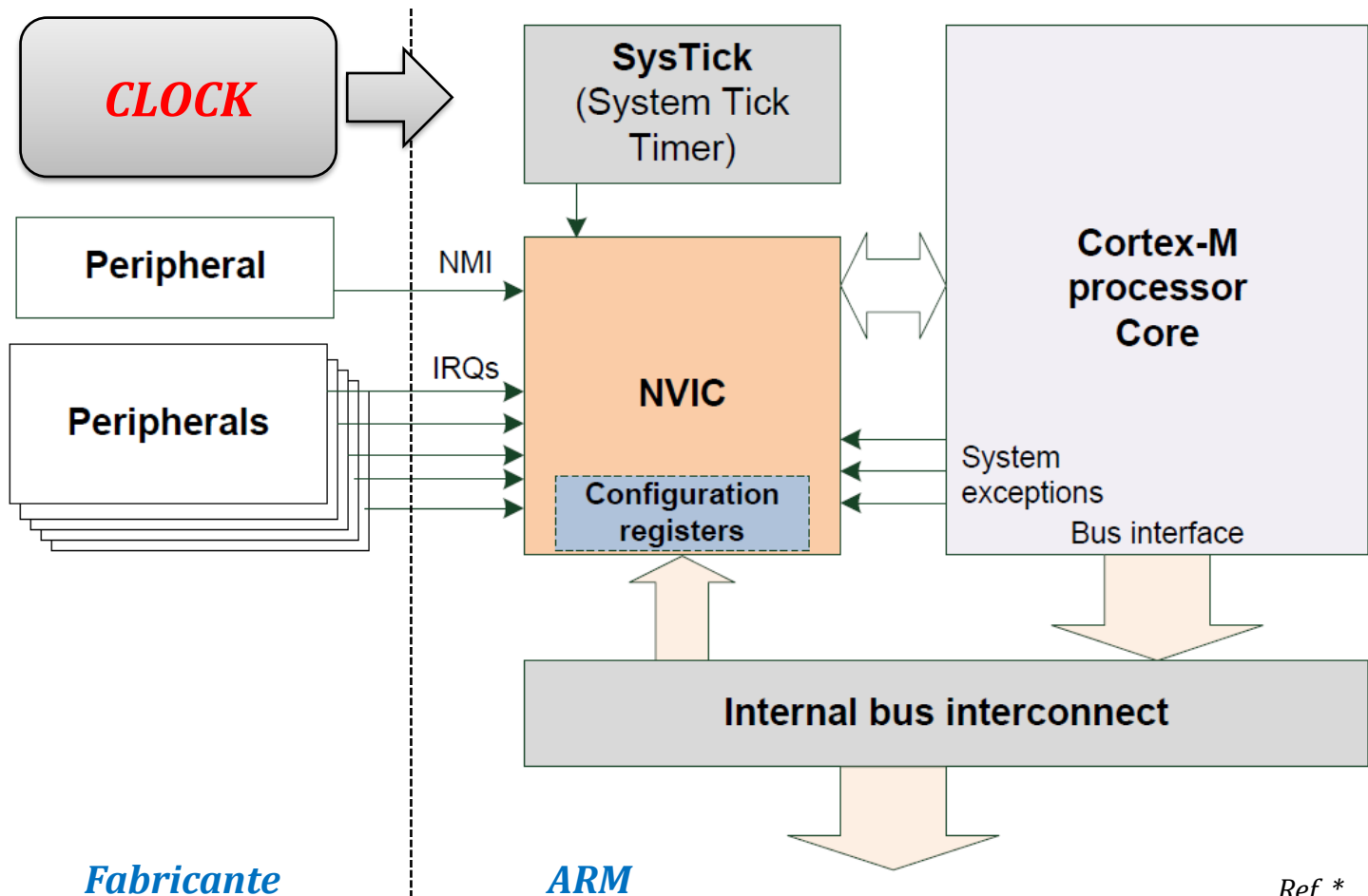
Prof: DaLuz



- Arquitetura
- RTOS
- Referências

Arquitetura:

Cortex-M – Conceção



Fabricante

ARM

*Ref. **



- Arquitetura
- Referências
- Referências

Arquitetura:

Clock (pg. 230): *Clock Tree* (pg. 233)

Interno:



Precision Internal Oscillator
(PIOSC) - POR - 16 MHz $\pm F_{PIOSC}$

Tabela: 27-19 pg 1837 - RC

Low-Frequency Internal Oscillator
(LFIOOSC) – 33 KHz

Hibernation Module RTC Oscillator
(RTCOSC / LFIOOSC) Clock Source

Externo:



Main Oscillator
(MOSC) - 5 MHz a 25 MHz

sem PLL de 4 MHz a 25 MHz – CMOS / XTAL

Hibernation Module RTC Oscillator
(RTCOSC) Clock Source
32.768 KHz



- Arquitetura
- RTOS
- Referências

Arquitetura:

Clock (pg. 230): *Clock Source* (pg. 231)

Clock Source	Drive PLL Capability?	PLL Enabled, RSCLKCFG Bit Encodings	SysClk generation capability?	SysClk generation enabled, RSCLKCFG Bit Encodings
Precision Internal Oscillator (PIOSC)	Yes	USEPLL = 1, PLLSRC = 0x0	Yes	USEPLL = 0, OSCSRC = 0x0
Main Oscillator (MOSC)	Yes	USEPLL = 1, PLLSRC = 0x3	Yes	USEPLL = 0, OSCSRC = 0x3
Low Frequency Internal Oscillator (LFIOSC)	No	-	Yes	USEPLL = 0, OSCSRC = 0x2
Hibernation Module RTC Oscillator (RTCOSC). 32.768-kHz Oscillator or HIB LFIOSC	No	-	Yes	USEPLL = 0, OSCSRC = 0x4



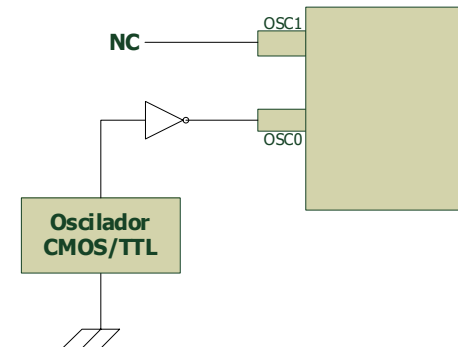
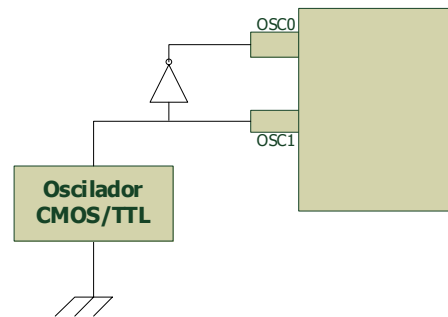
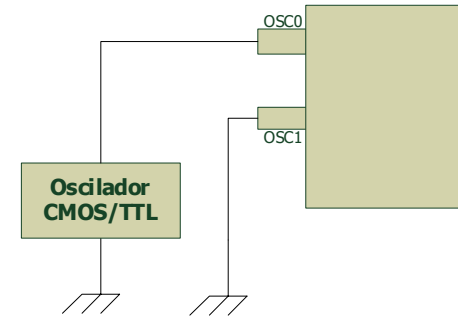
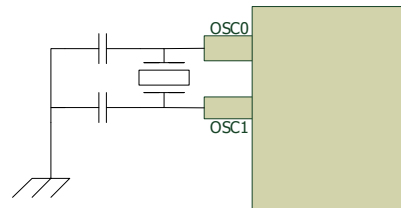
- Arquitetura

- RTOS

- Referências

Arquitetura:

Clock (pg. 230): *Clock Source* (pg. 231)





- Arquitetura
- RTOS
- Referências

Arquitetura:

Clock : Observações

ADC

ADC *Digital Block* utiliza o **system clock**.

ADC *Analog Block* utiliza um **clock** separado (**ADC Clock**).

ADC Clock pode vir do **PLL**, **PIOSC** ou **MOSC**.

O **clock** deve ser no mínimo de **16 MHz**.

USB

PLL é utilizado para prover **60MHz**, para conversão serial.

O **clock** do sistema deve ser no mínimo **30 MHz**.

Ethernet

Ethernet MAC é conectado ao **system clock**.

Ethernet PHY requer **25 MHz** e o sinal pode ser do **MOSC**.

Outros

UART, **SSI** e **Clock control Registers** usam o **system clock**.

PIOSC também pode ser utilizado.



- Arquitetura
- RTOS
- Referências

Arquitetura:

PLL: Equações de Configuração

$$\text{Input Freq. PLL} = f_{in} = \frac{f_{XTAL}}{(N + 1)} \text{ ou } \frac{f_{PIOSC}}{(N + 1)}, \text{ onde } f_{in} \in (5 \text{ MHz a } 25 \text{ MHz})$$

$$\text{Output Freq. PLL} = f_{VCO} = f_{in} \times \mathbf{MDIV}, \text{ onde } f_{VCO} \in (250 \text{ MHz a } 500 \text{ MHz})$$

$$\mathbf{MDIV} = \mathbf{MINT} + \frac{\mathbf{MFRAC}}{1024}, \text{ onde } \mathbf{MFRAC} \text{ deveria ser sempre } 0, \text{ menor } \mathbf{Jitter}$$

$$\text{SysClk} = \frac{f_{VCO}}{(Q + 1) \cdot (\mathbf{PSYSDIV} + 1)},$$

As variáveis Q, N, \mathbf{MINT} e \mathbf{MFRAC} estão no registrador $\mathbf{PLLREQn}$.
E o fator de divisão do systema está no registrador $\mathbf{RCLKCFG}$.



- Arquitetura
- RTOS
- Referências

Arquitetura:

Registradores do *System Control*

Item 5.4 do datasheet, pg. 247:

http://www.elf74.daeln.com.br/Pdfs/Datasheet_TM4C1294NCPDT_v1.01.pdf



- Arquitetura

- RTOS

- Referências

Arquitetura:

Registradores do *System Control*

MOSCCTL: Main Oscillator Control 0x400FE07C - pg. 273															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
reservado								reservado							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reservado											O S C R N G	P W R D N	N O X T A L	M O S C I M	C V A L

OSCRNG <=> *Oscillator Range*

OSCRNG = **0x0** => Low Frequency Range

= **0x1** => High Frequency Range (equal to or greater than 10 MHz)

PWRDN <=> *Power Down*

PWRDN = **0x0** => Power to main oscillator circuit is enabled

= **0x1** => Main Oscillator circuit is powered down

NOXTAL <=> *No MOSC/Crystal Connected*

NOXTAL = **0x0** => xtal presente

= **0x1** => xtal ausente

CVAL <=> *Clock Validation for MOSC*

CVAL = **0x0** => The MOSC monitor circuit is disabled

= **0x1** => The MOSC monitor circuit is enabled



- Arquitetura

- RTOS

- Referências

Arquitetura:

Registradores do *System Control*

DIVSCLK: Divisor and Source Clock Configuration 0x400FE148 - pg. 284

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN	reservado													SRC	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reservado								DIV							

EN \Leftrightarrow *DIVSCLK Enable*

EN = **0x0** \Rightarrow The clock output is disabled

= **0x1** \Rightarrow Clock output is enabled

SRC \Leftrightarrow *Clock Source*

SRC = **0x0** \Rightarrow System Clock

= **0x1** \Rightarrow PIOSC

= **0x2** \Rightarrow MOSC

= **0x3** \Rightarrow Reservado

DIV \Leftrightarrow *Divisor Value*

SRC = **0x0** \Rightarrow Divided by 1 (N+1)

= **0x1** \Rightarrow Divided by 2 (N+1)

= ... \Rightarrow

= **0xN** \Rightarrow Divided by (N+1)



- Arquitetura
- RTOS
- Referências

Arquitetura:

Registradores do *System Control*

RSCLKCFG: Run and Sleep Mode Configuration Register 0x400F.EB00 - pg. 275															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M E M T I M U	N E W F R E Q	A C G	U S E P L L	PLLSRC				OSCSRC				OSYSDIV			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSYSDIV						PSYSDIV									

MEMTIMU <=> *Memory Timing Register Update*

MEMTIMU = **0x0** => This bit is automatically cleared by hardware
 = **0x1** => Setting this bit causes the *MEMTIM0* register value to be applied, and the memory timing to be updated

NEWFREQ <=> *DIVSCLK Enable*

NEWFREQ = **0x0** => This bit is automatically cleared by hardware
 = **0x1** => This bit controls the activation of the values in the **PLLFREQ0** and **PLLFREQ1** registers as applied to the **PLL**

ACG <=> *Auto Clock Gating*

ACG = **0x0** => The Run-Mode Clock Gating Control (*RCGCn*) registers are used when the microcontroller enters a sleep mode
 = **0x1** => If the microcontroller is in sleep mode, the *SCGCn* registers are used to control the clocks distributed to the peripherals



- Arquitetura

- RTOS

- Referências

Arquitetura:

Registradores do *System Control*

USEPLL <=> Use PLL

USEPLL = **0x0** => Clock source specified by *OSCSRC* field

= **0x1** => Clock source specified by the PLL

PLLSRC <=> PLL Source

PLLSRC = **0x0** => *PIOSC* is the PLL input clock source

= **0x3** => *MOSC* is the PLL input clock source

= **0xN** => Reservado

OSCSRC <=> Oscillator Source

OSCSRC = **0x0** => *PIOSC* is oscillator source

= **0x2** => *LFIOSC* is oscillator source

= **0x3** => *MOSC* is oscillator source

= **0x4** => Hibernation Module RTC Oscillator (*RTCOSC*)

= **0xN** => Reservado

OSYSDIV <=> Oscillator System Clock Divisor

OSYSDIV = **0xN** => The divisor value is the *OSYSDIV* field value + 1

$fsyclk = foscclk / (OSYSDIV + 1)$

PSYSDIV <=> PLL System Clock Divisor

PSYSDIV = **0xN** => The divisor value is the *PSYSDIV* field value + 1

$fsyclk = fVCO / (PSYSDIV + 1)$



- Arquitetura

- RTOS

- Referências

Arquitetura:

Registradores do *System Control*

PLLFREQ0: PLL Frequency 0 0x400F.E160 - pg. 292															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
reservado								P L L P W R	reservado			MFRAC			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MFRAC						MINT									

$$fVCO = (fIN * MDIV)$$

$$fIN = fXTAL / (Q+1) (N+1) \text{ or } fPIOSC / (Q+1) (N+1)$$

$$MDIV = MINT + (MFRAC / 1024)$$

USEPLL = **0x1** => This bit controls power to the PLL

MFRAC = **MFRAC** value

MINT = **MINT** value



- Arquitetura
- RTOS
- Referências

Arquitetura:

Registradores do *System Control*

PLLFREQ1: PLL Frequency 1 0x400F.E164 - pg. 293															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
reservado															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reservado			Q					reservado			N				

$$fVCO = (fIN * MDIV)$$

$$fIN = fXTAL / (Q+1) (N+1) \text{ or } fPIOSC / (Q+1) (N+1)$$

$$MDIV = MINT + (MFRAC / 1024)$$

Q = PLL **Q** value

N = PLL **N** value



- Arquitetura

- RTOS

- Referências

Arquitetura:

Registradores do *System Control*

ALTCLKCFG - *Alternate Clock Configuration* - pg. 280

DSCLKCFG - *Deep Sleep Clock Configuration Register* - pg. 281

PIOSCCAL - *Precision Internal Oscillator Calibration* - pg. 289

PIOSCSTAT - *Precision Internal Oscillator Statistics* - pg. 291

PLLSTAT - *PLL Status* - pg. 294

SLPPWRCFG - *Sleep Power Configuration* - pg. 295

DSLPPWRCFG - *Deep-Sleep Power Configuration* - pg. 297

PLLSTAT - *PLL Status* - pg. 294



- Arquitetura
- RTOS
- Referências

Arquitetura:

Exercício:

Abrir o arquivo: **sysctl.c**
Localizar a função: **SysCtlClockFreqSet** – linha 2126
Dar uma olhada geral neste arquivo ...
`\TivaWare_C_Series-2.2.-.295\driverlib\sysctl.c`



- Arquitetura
- RTOS
- Referências

RTOS

Concorrência / Paralelismo



Ref. *



- Arquitetura
- RTOS
- Referências

RTOS

Concorrência / Paralelismo

1 x Programa:

Quanto maior o programa mais difícil é a construção do P.P.

Programação convencional pode gerar problemas temporais

N x Tarefas

*Menores tarefas
Dividir o P.P. em partes*

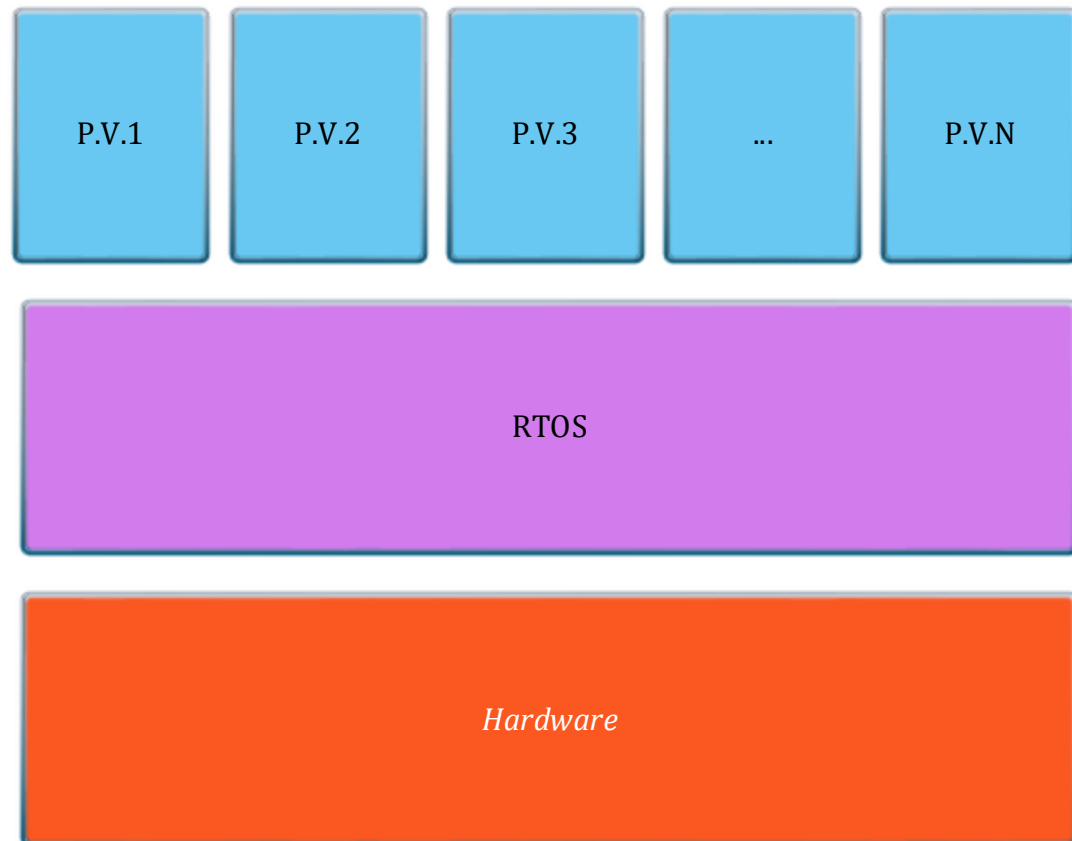
Depois basta rodar todas elas em paralelo



- Arquitetura
- RTOS
- Referências

RTOS

Hardware Single Core

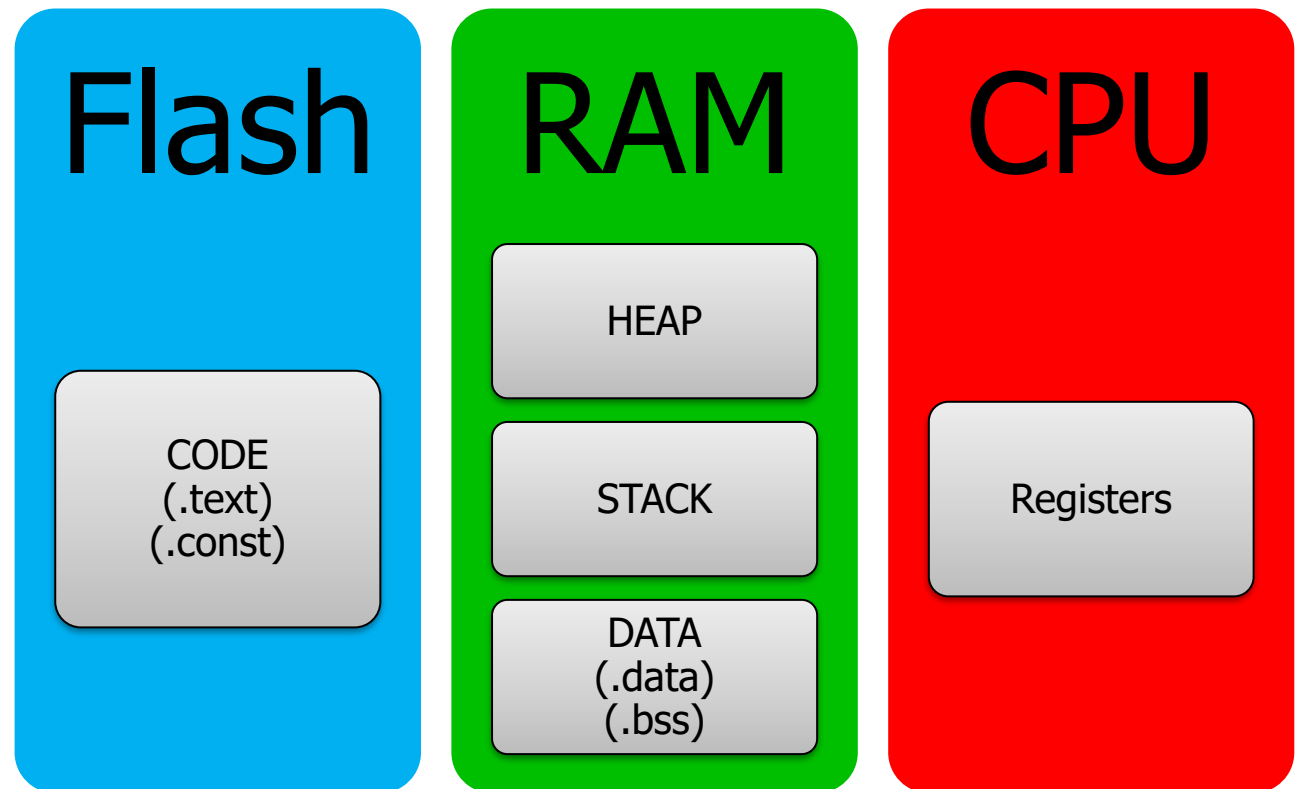




- Arquitetura
- RTOS
- Referências

RTOS

Hardware Single Core

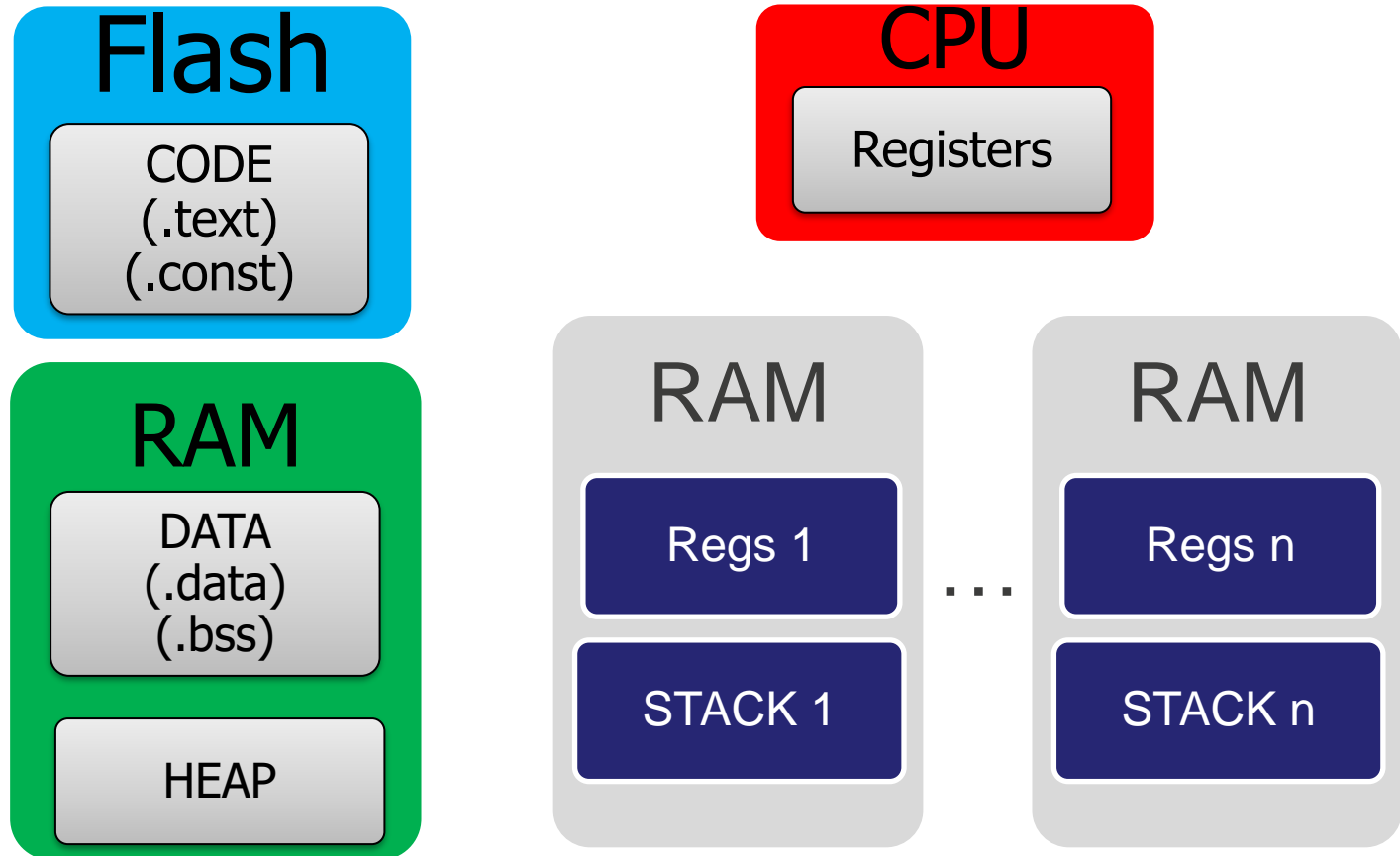




- Arquitetura
- RTOS
- Referências

RTOS

Hardware Single Core





- Arquitetura
- RTOS
- Referências

RTOS

Concorrência / Paralelismo

Thread:

Parte de um Programa / APP.

Task é um conceito lógico

Processo

*MMU
Programa / APP.*

*Cada processo pode ter diversas
Threads*



- Arquitetura
- RTOS
- Referências

RTOS

Concorrência / Paralelismo

Ñ Preemptivo:

Depende da Colaboração

O S.O. libera o controle para a Thread

Preemptivo

O S.O. gerência tudo

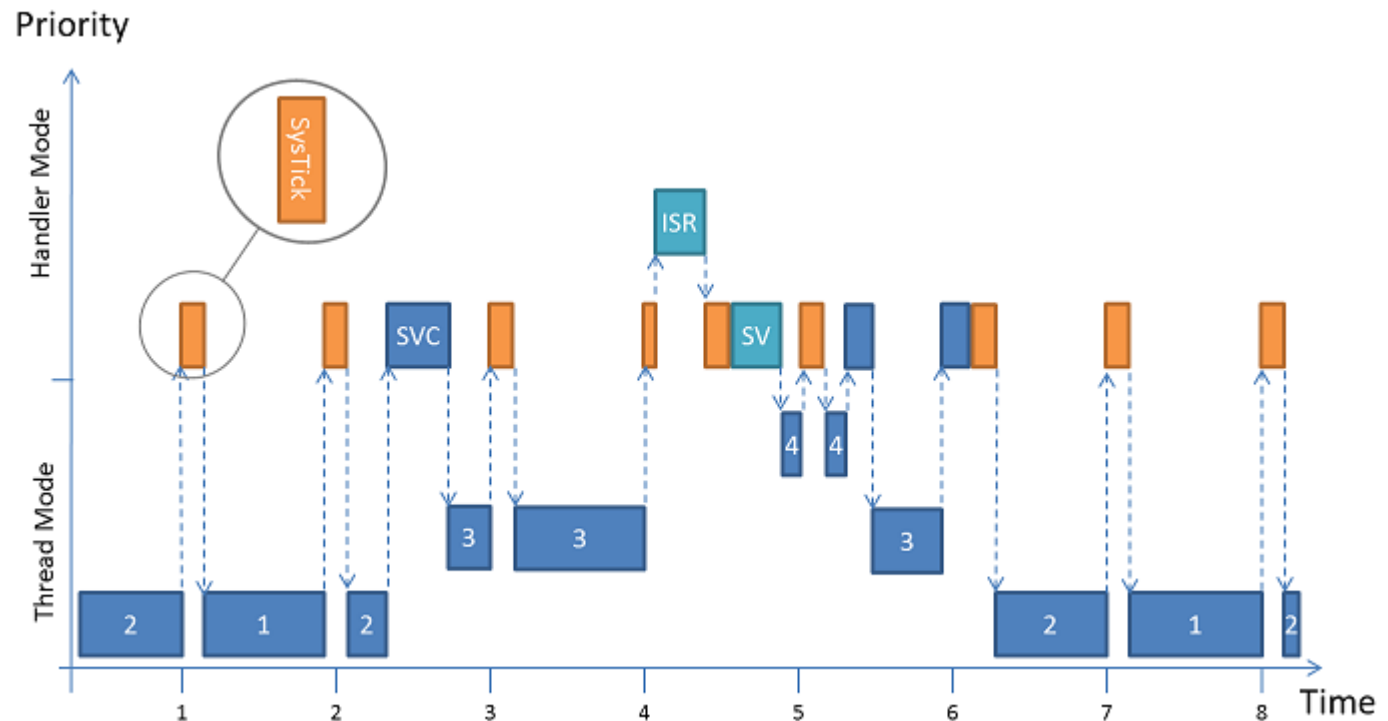
O S.O. aloca e desaloca os recursos para as Threads.



- Arquitetura
- RTOS
- Referências

RTOS

Prioridade:





- Arquitetura
- RTOS
- Referências

RTOS

Ressalvas:

Interdenpendentes:

*Caso em que as tarefas não
compartilham nada.*

*Nestes casos basta um
Gerenciamento simples de uso*

Dependências:

Gerência necessita de cuidados

*Podem gerar DeadLocks,
Inversão de Prioridade ou
Starvation*



- Arquitetura
- RTOS
- Referências

RTOS

Questões Importantes

Comunicação entre as tarefas

Sincronismo entre as tarefas

*Compartilhamento de recursos
Exclusão mútua*

Agendamento de tarefas



- Arquitetura
- RTOS
- **Referências**

Referências:

Continuação dos Labs 1 a 4:

- ▣ * Refs ↔ Renesas.com, Pixabay.com, wikimedia.org, flickr, community.arm.com, Undergraduated course Renesas / CWS71-Sistemas Embarcados (Prof. Douglas P. B. Renaux e Robson Linhares), ytchannel Gustavo W. Dernardin, *ARMv7-M Architecture Reference Manual*, *CSW40-Sistemas Microcontrolados* – Prof. Peron, toshiba.semicon-storage.com, microncontrollerslab.com, lfelectronics.com.br