



EA871

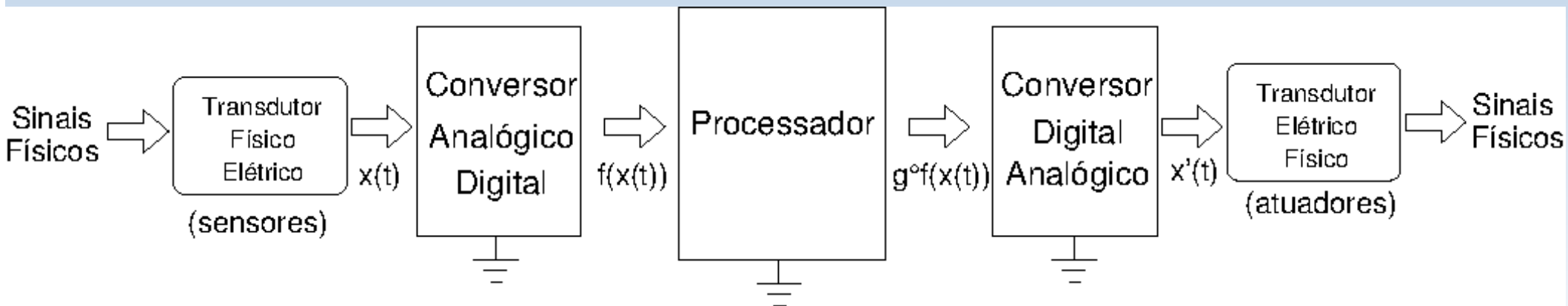
Conversores Analógico-Digital

Conversão disparada por *software*

Wu Shin – Ting
DCA – FEEC - Unicamp
Segundo Semestre de 2020

Conceitos

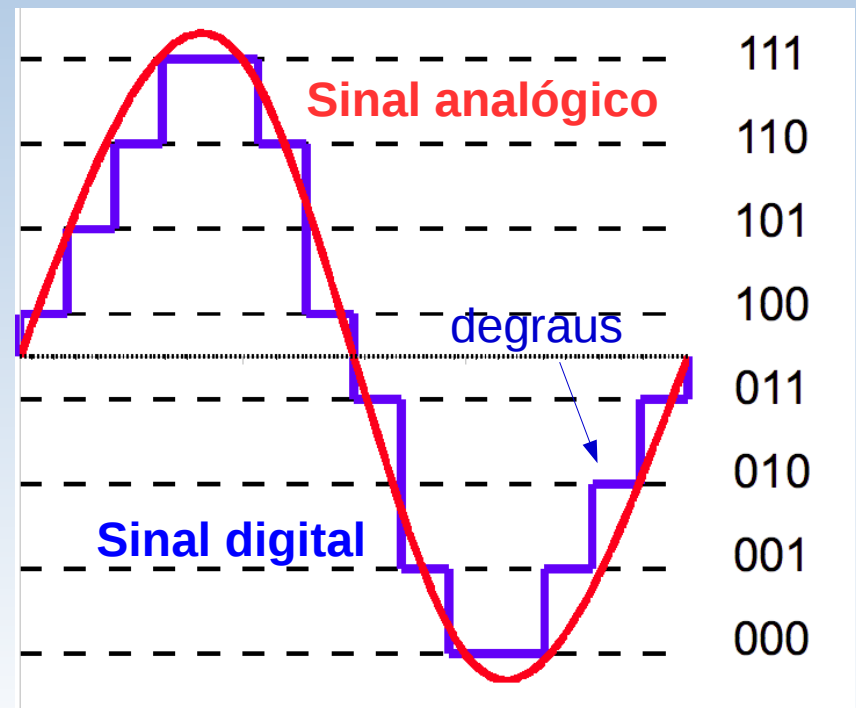
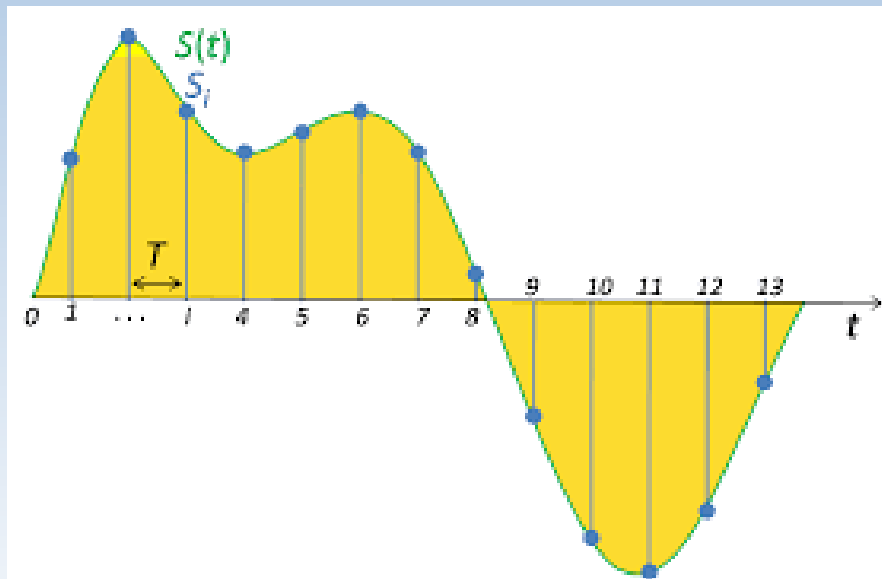
- **Transdutores:** dispositivos capazes de transformar a energia de uma natureza para outra.
- **Conversor Analógico-Digital:** circuito capaz de converter sinais analógicos em sinais digitais.
- **Conversor Digital-Analógico:** circuito capaz de converter sinais digitais em sinais analógicos.



Conceitos

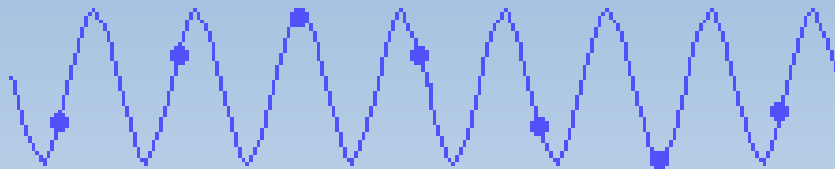
Conversão Analógica-Digital

- **Amostragem:** seleção de amostras de um sinal analógico.
- **Quantização:** aproximação dos valores das amostras em valores discretos representáveis num dispositivo digital.

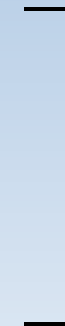
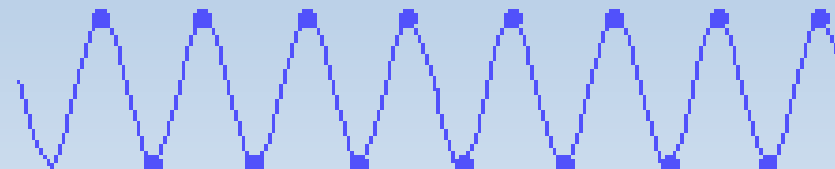


Conceitos

- **Teorema de Amostragem Nyquist-Shannon:** um sinal pode ser perfeitamente reconstruído se a frequência de amostragem for maior que 2 vezes a maior frequência do sinal.

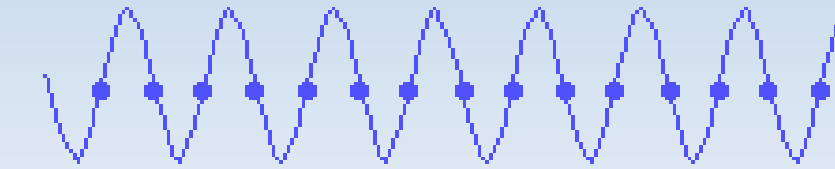


$$f_s < 2f$$

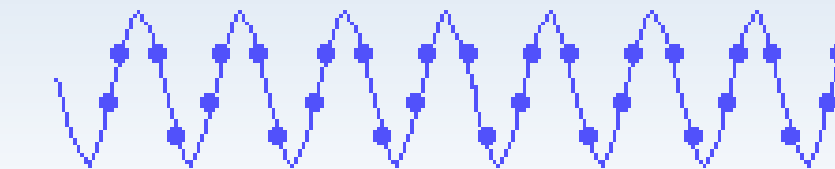


$$f_s = 2f$$

Frequência de Nyquist



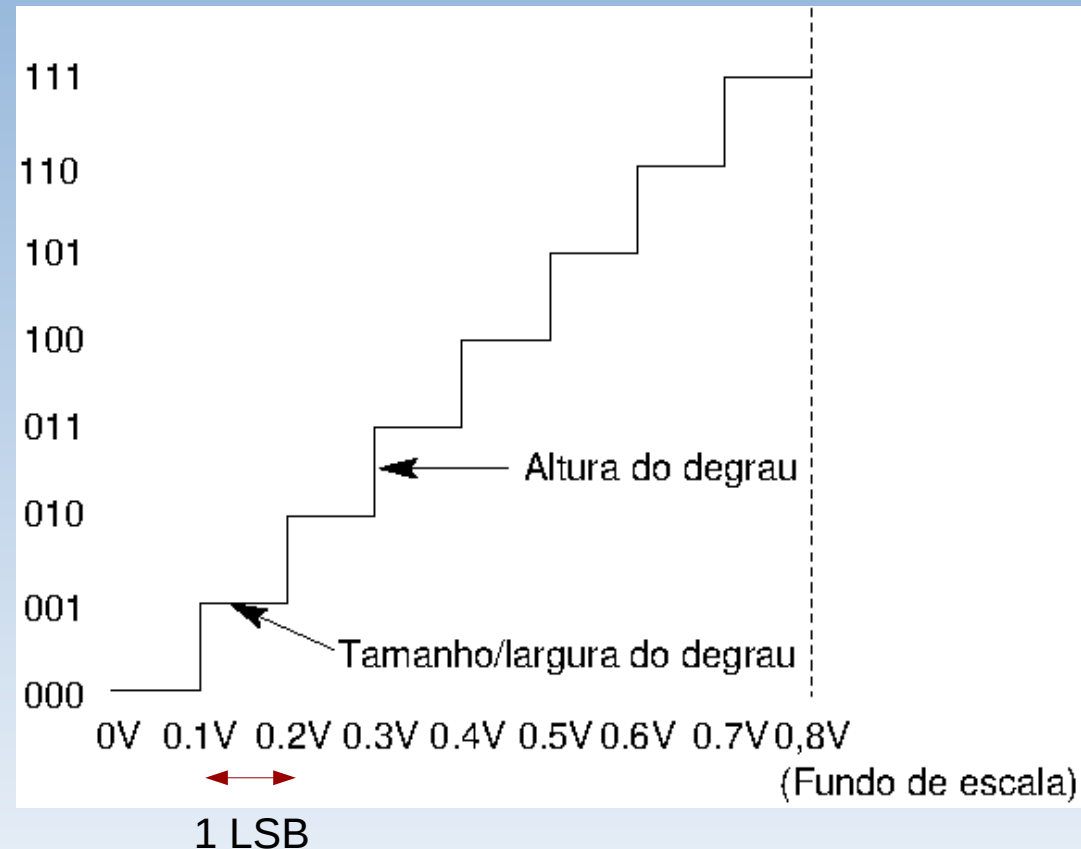
$$f_s > 2f$$



Conceitos

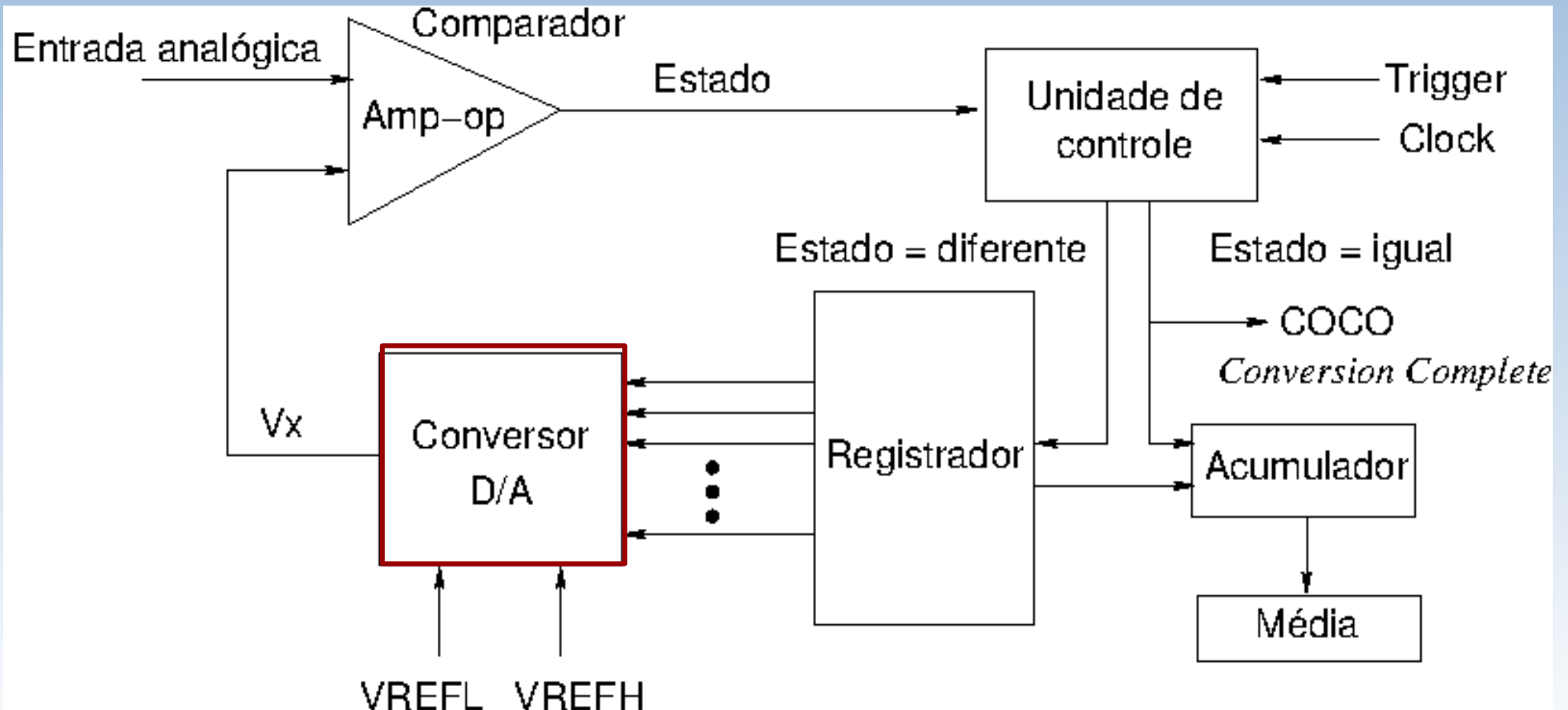
- **Fundo de escala:** maior valor analógico.
- **Níveis de resolução:** valores discretos representáveis.
 $N \text{ bits} \rightarrow 2^N \text{ níveis.}$
- **Resolução** (tamanho de degrau): diferença entre dois valores analógicos representados pelos dois códigos binários adjacentes. É especificada em quantidade de *bits* (N).

$$1 \text{ LSB} = \frac{FS}{\text{Níveis de resolução}} = \frac{FS}{2^N}$$



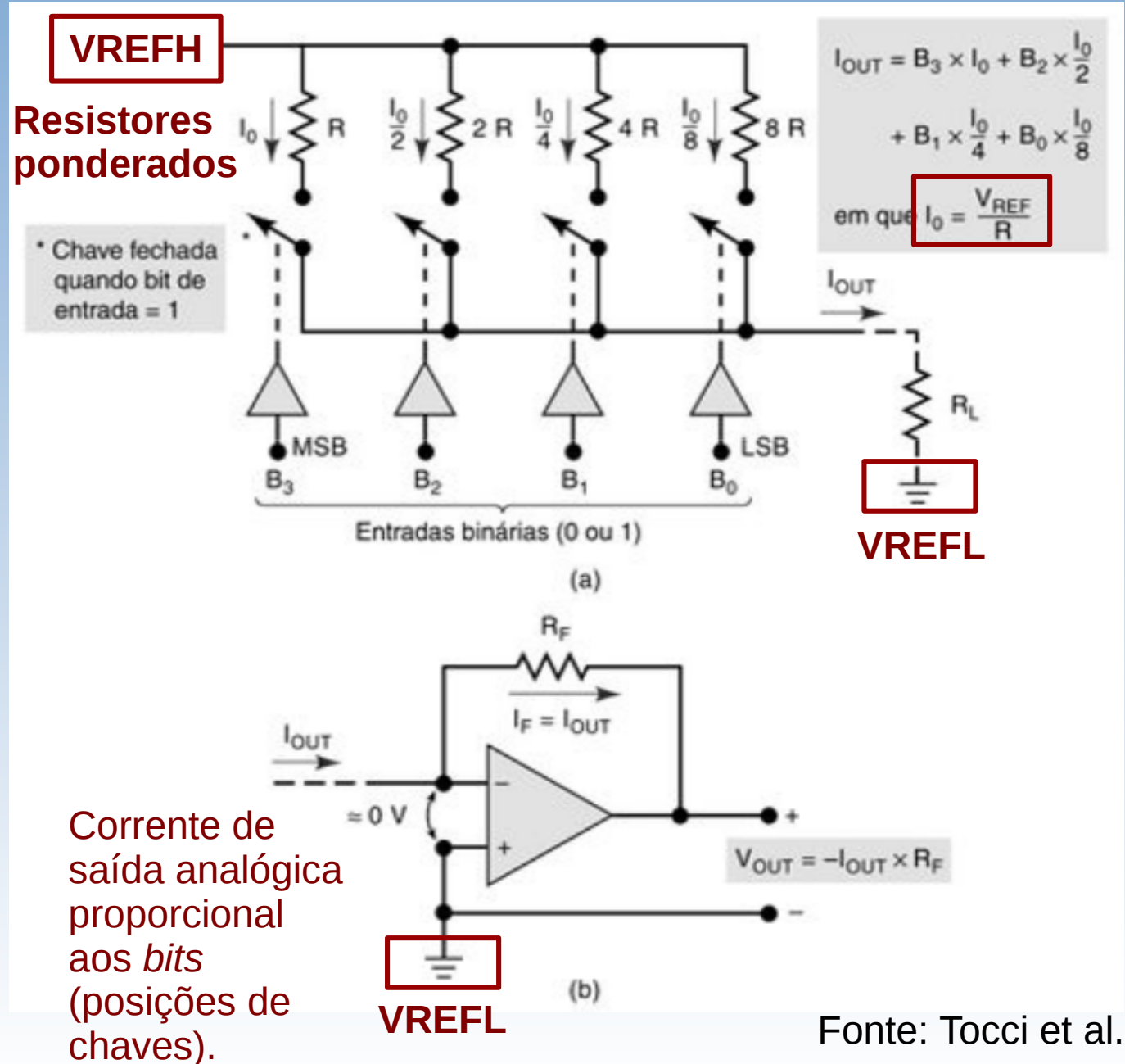
Conceitos

- **Conversor analógico-digital:** circuito que tem como entrada uma tensão analógica e gera na saída um código binário (sinal digital) que representa a entrada analógica.



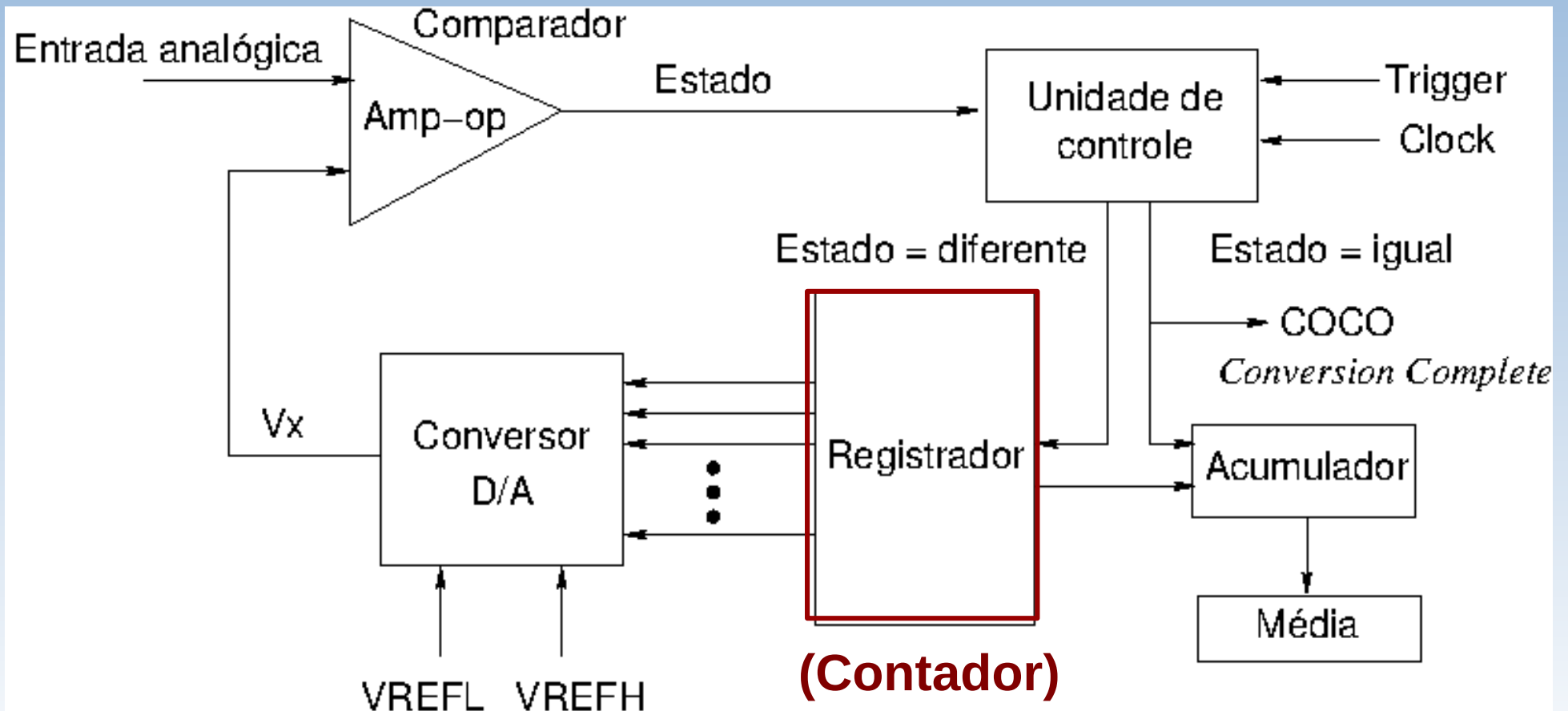
Conceitos

- **Conversor digital-analógico:** circuito que tem como entrada um código binário e gera na saída uma tensão ou uma corrente proporcional ao valor do código binário.



Conceitos

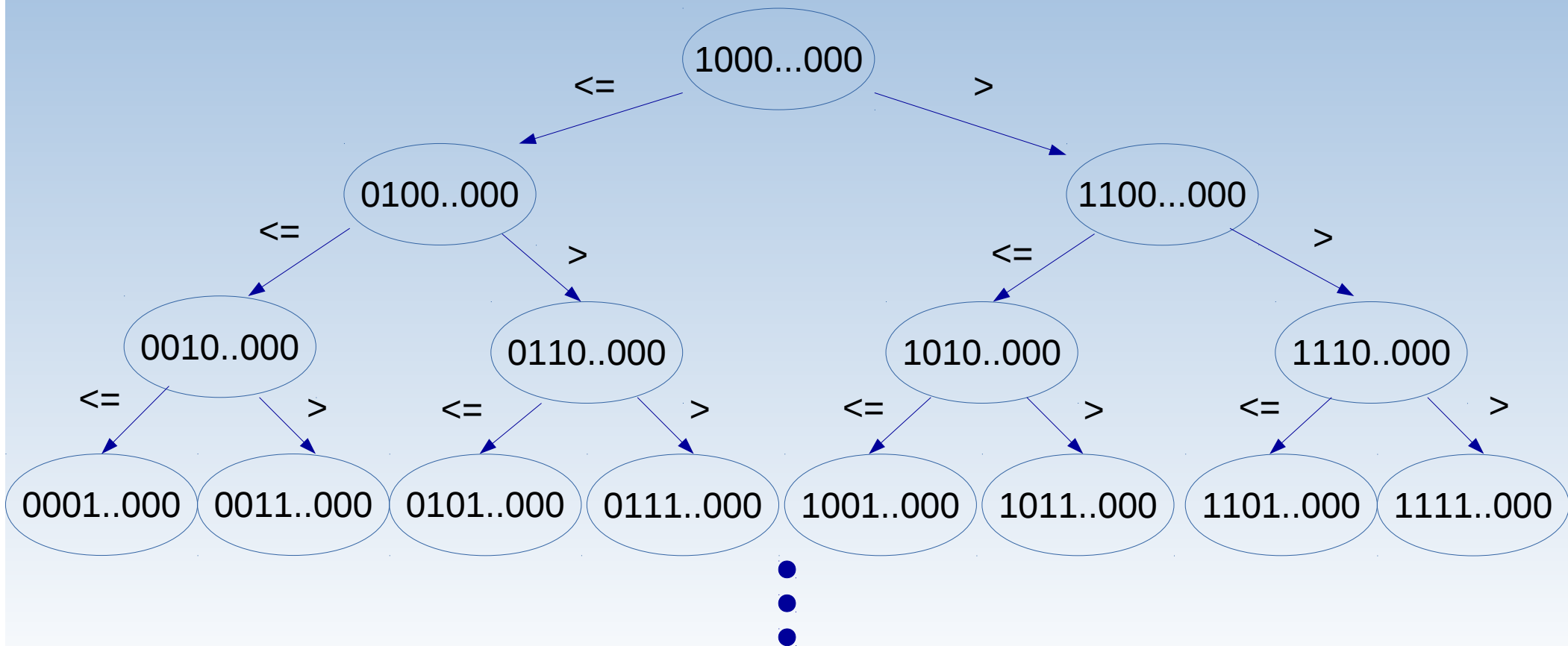
- **Registrador** (de códigos binários): circuito responsável pela geração de todos os possíveis códigos binários.



N bits $\rightarrow 2^N$ comparações (pior caso)

Conceitos

- **Registrador de Aproximações Sucessivas:** geração sucessiva de códigos binários que reduz a quantidade de comparações para $(\log_2 N + 1)$ em N bits no pior caso.



Conceitos

- **Linearidade:** a relação linear entre tensão e valor binário é observada para todos os valores $V_{REFL} \leq V_{entrada} \leq V_{REFH}$
- **Monotonicidade:** Entrada A > Entrada B → Código A' > Código B'.
- **Tensões de referência:** são os limitantes dos valores de (tensões de) entrada processáveis por um conversor AD de N bits.

$$Código = (Entrada Analógica - V_{REFL}) \frac{2^N}{V_{REFH} - V_{REFL}}$$

- $Entrada Analógica < V_{REFL}$: usualmente Código = 0.
- $V_{REFH} < Entrada Analógica$: usualmente Código = $2^N - 1$.

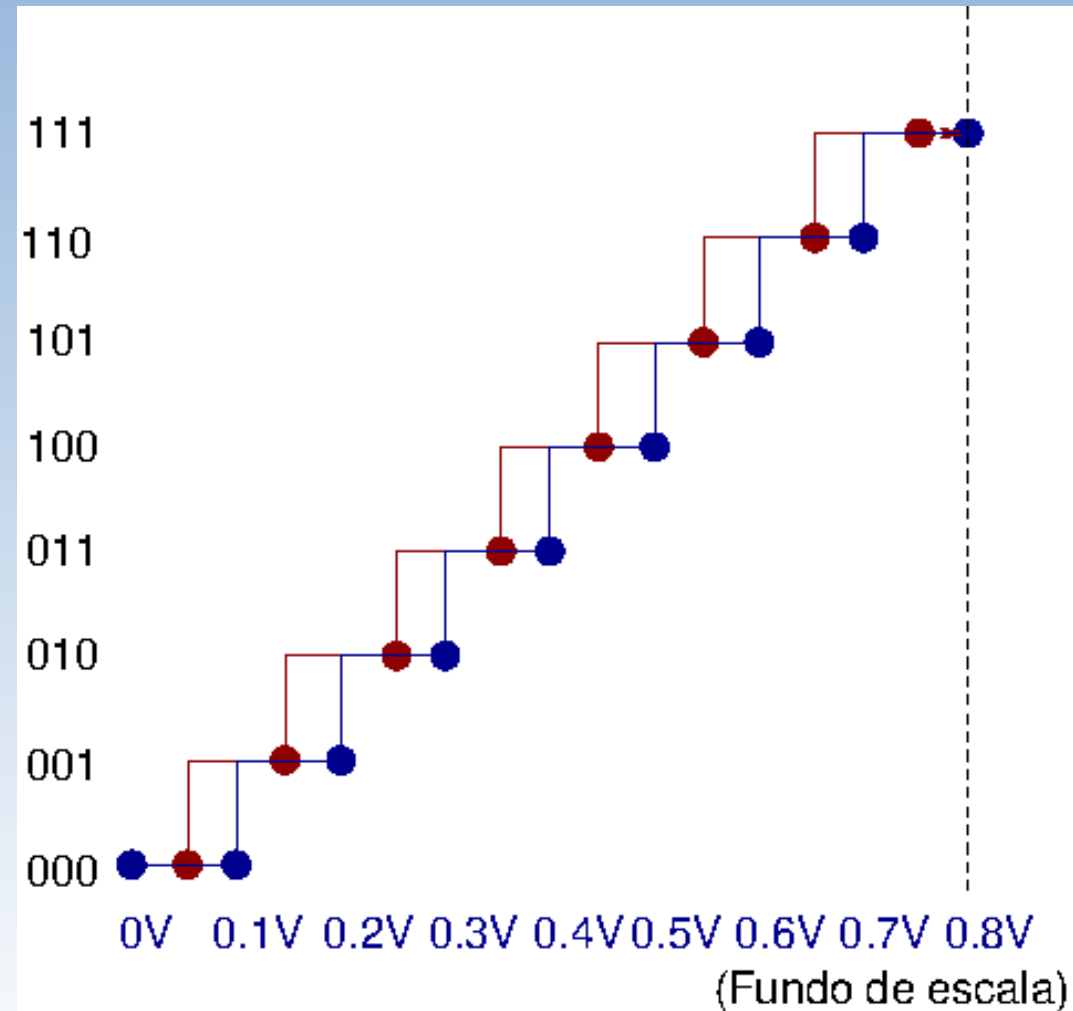
•

Conceitos

- **Tempo de conversão:** intervalo de tempo entre o *trigger* de inicialização e a ativação do *flag* COCO (*Complete Conversion*). Neste intervalo o sinal de entrada deve ser mantido constante.
 - **Tempo de amostragem:** intervalo de tempo necessário para amostrar o valor do sinal de entrada.
 - **Tempo de conversão:** intervalo de tempo necessário para transformar o valor amostrado num código binário.

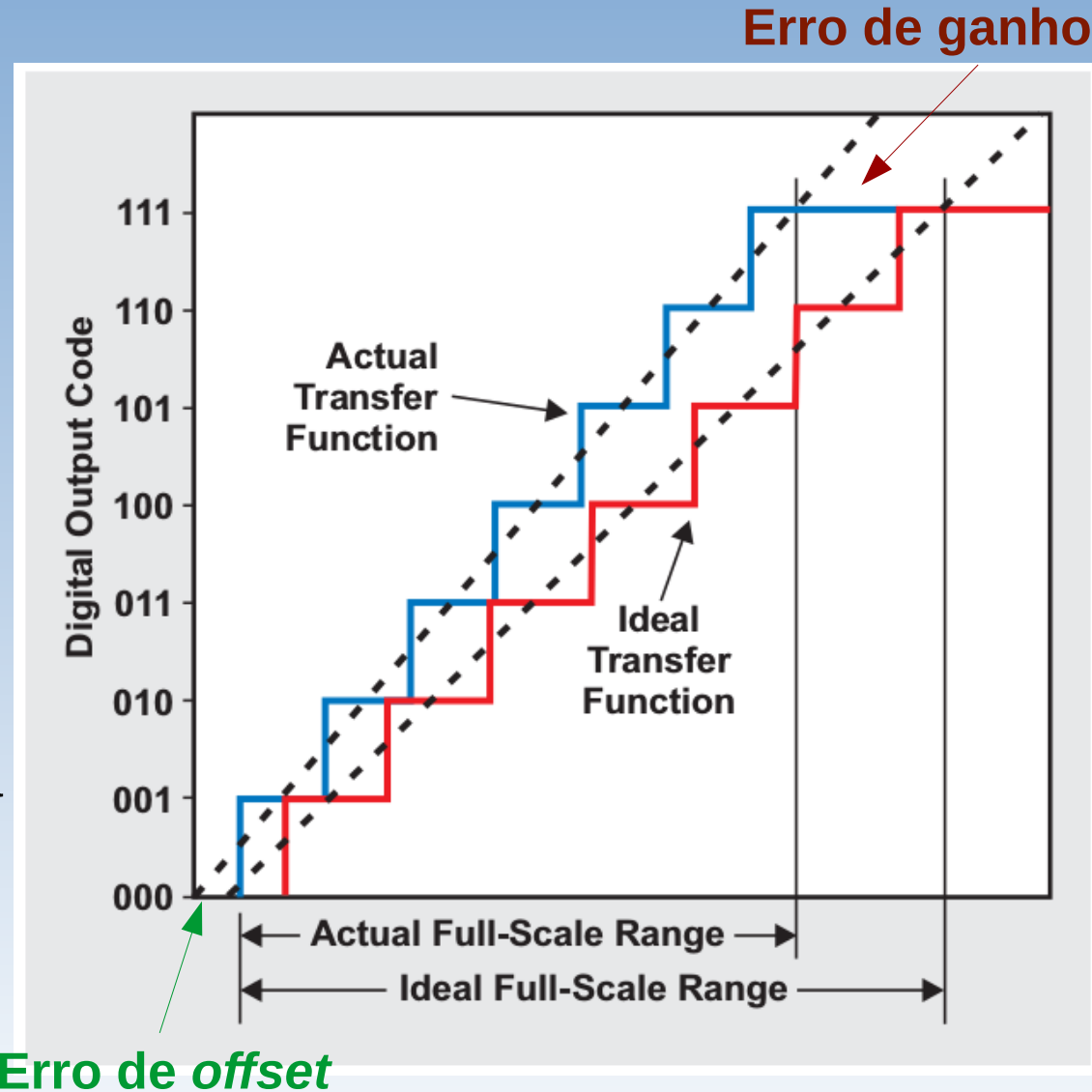
Conceitos

- **Transições de códigos binários:**
 - no meio do degrau
 - no fim do degrau
- **Erro de quantização** (erro de arredondamento): diferença entre o valor do sinal analógico amostrado e o valor quantizado.
 - no meio do degrau: $\pm \frac{1}{2} LSB$
 - no fim do degrau: $-LSB$



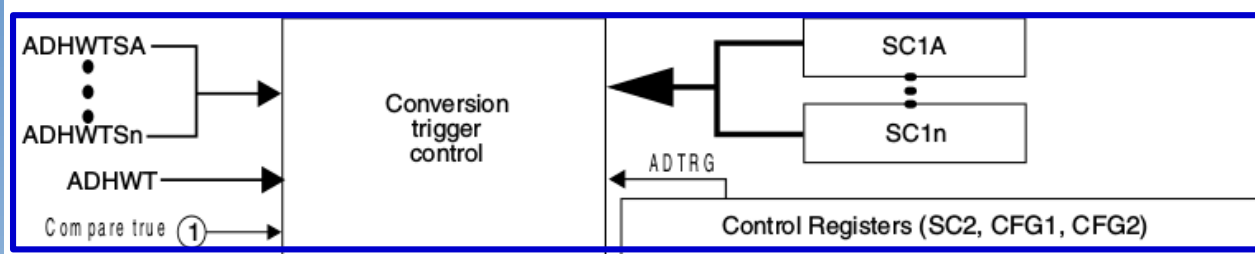
Conceitos

- **Erro de *offset***: o valor de entrada para o qual o código binário de saída seja 0.
- **Erro de *ganho***: a diferença entre o valor real e o valor ideal no fundo de escala.
- **Não-linearidade diferencial**: diferença nas larguras dos degraus em relação à largura ideal.
- **Não-linearidade integral**: diferenças entre os valores de saída e os valores ideais.
- **Calibração**: comparação dos valores medidos com os valores padrão.

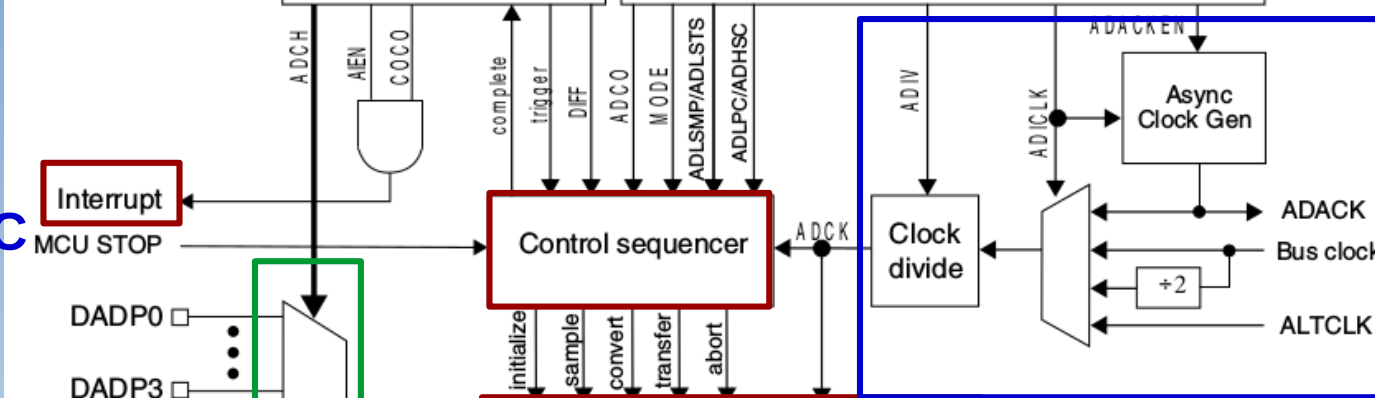


Módulo ADC

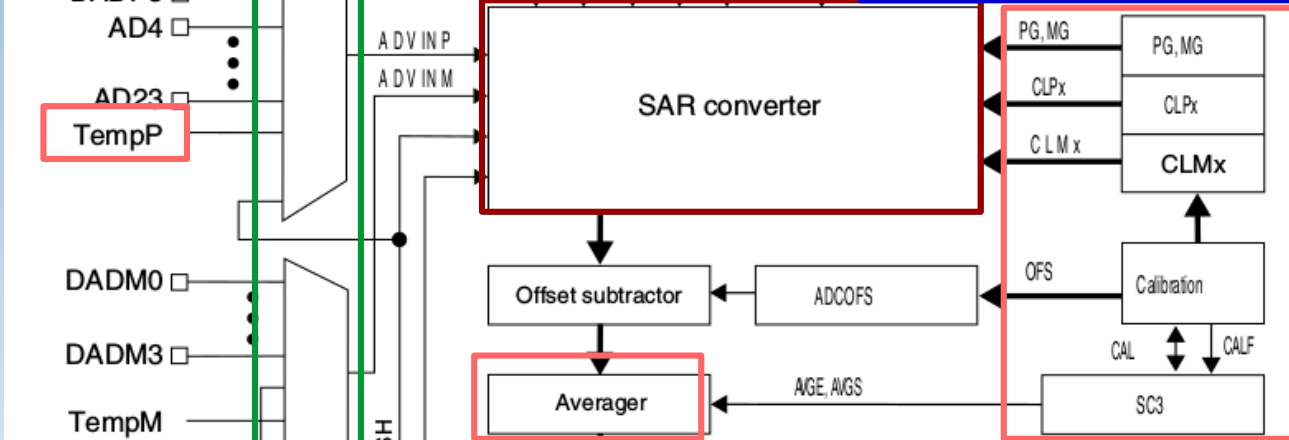
- Conversor analógico-digital de **aproximações sucessivas** de **16 bits**, com 4 pares de entradas diferenciais (bipolares) e **24 entradas unipolares multiplexáveis**.
 - Modos de saída para entradas unipolares: 16-, 12-, 10- e 8-*bits*.
 - *Trigger* por *software* ou *hardware*
 - Modo de auto-calibração
 - Diferentes velocidades de amostragem
 - Função de valor médio das amostras
 - *Conversão única ou contínua*.
 - Função de comparação
 - Sensor de temperatura AN3031 conectado ao módulo (AD26)



trigger



clock



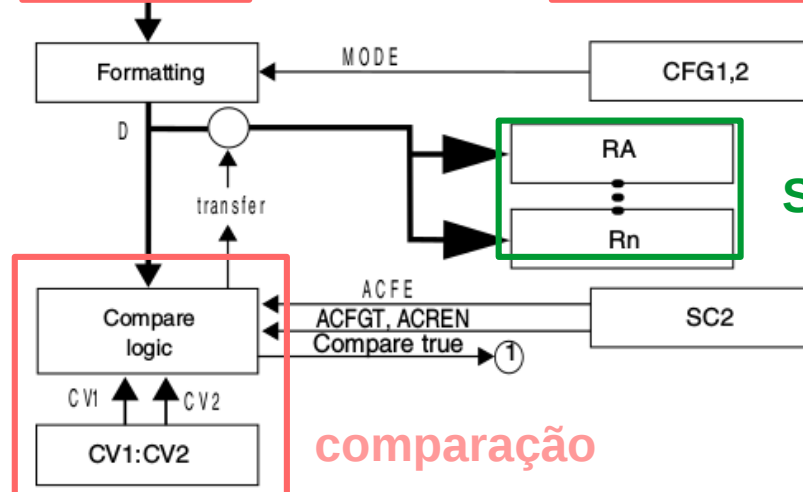
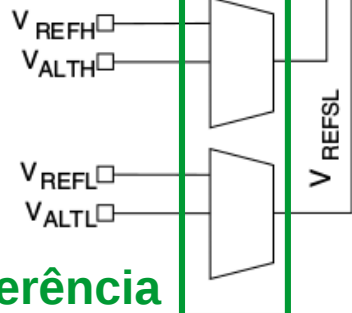
calibração

Entradas analógicas

3000mV

0mV

Tensões de referência



Saída digital

comparação

NVIC IRQ15
0x0000_007C

Interrupt

MCU STOP

TempP

Averager

RA

Rn

Compare logic

CV1:CV2

Entradas Analógicas

3.7.1.3.1 ADC0 Channel Assignment

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
00000	DAD0	ADC0_DP0 and ADC0_DM0	ADC0_DP0/ADC0_SE0
00001	DAD1	ADC0_DP1 and ADC0_DM1	ADC0_DP1/ADC0_SE1
00010	DAD2	ADC0_DP2 and ADC0_DM2	ADC0_DP2/ADC0_SE2
00011	DAD3	ADC0_DP3 and ADC0_DM3	ADC0_DP3/ADC0_SE3
00100 ¹	AD4a	Reserved	ADC0_DM0/ADC0_SE4a
00101 ¹	AD5a	Reserved	ADC0_DM1/ADC0_SE5a
00110 ¹	AD6a	Reserved	ADC0_DM2/ADC0_SE6a
00111 ¹	AD7a	Reserved	ADC0_DM3/ADC0_SE7a
00100 ¹	AD4b	Reserved	ADC0_SE4b
00101 ¹	AD5b	Reserved	ADC0_SE5b
00110 ¹	AD6b	Reserved	ADC0_SE6b
00111 ¹	AD7b	Reserved	ADC0_SE7b
01000	AD8	Reserved	ADC0_SE8
01001	AD9	Reserved	ADC0_SE9
01010	AD10	Reserved	Reserved

Entradas Analógicas

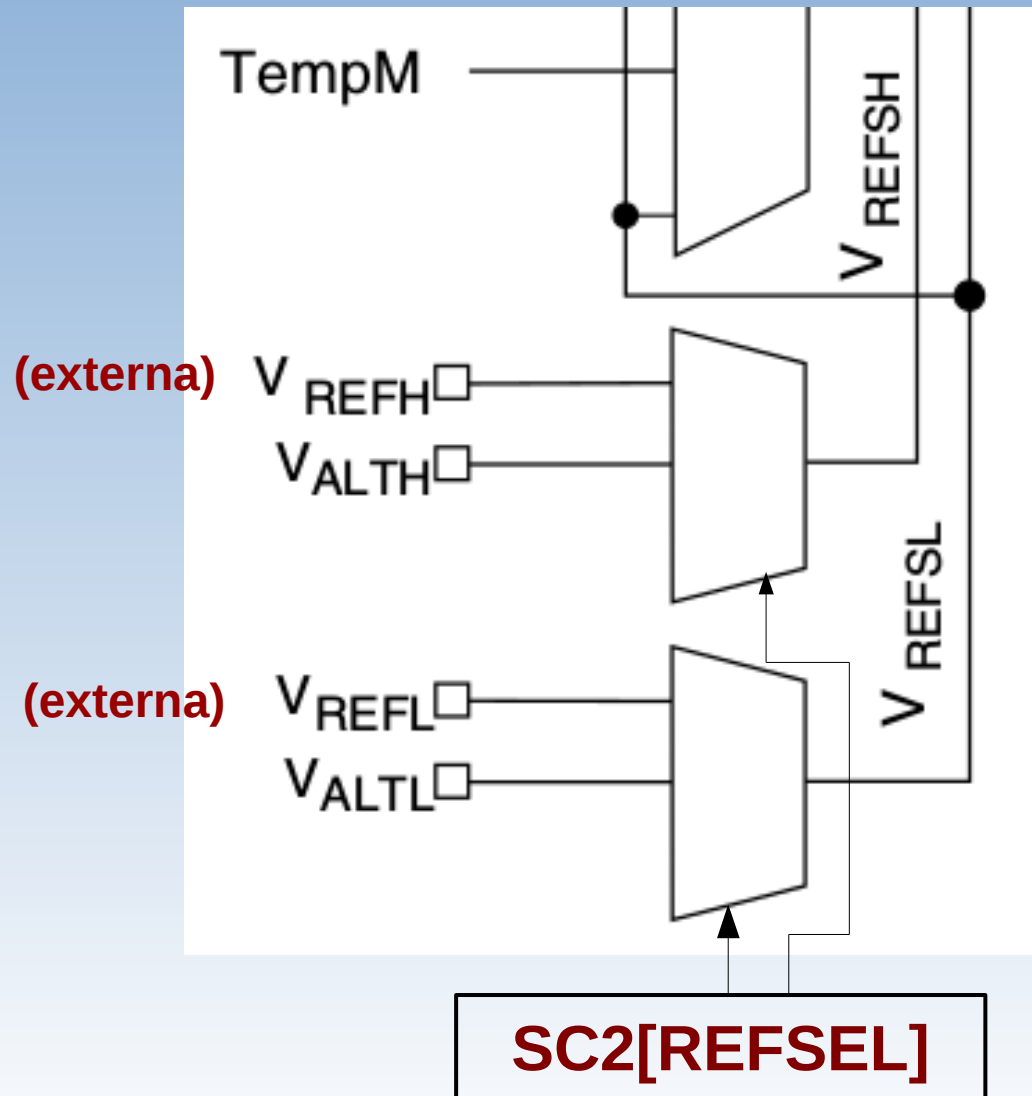
ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
01011	AD11	Reserved	ADC0_SE11
01100	AD12	Reserved	ADC0_SE12
01101	AD13	Reserved	ADC0_SE13
01110	AD14	Reserved	ADC0_SE14
01111	AD15	Reserved	ADC0_SE15
10000	AD16	Reserved	Reserved
10001	AD17	Reserved	Reserved
10010	AD18	Reserved	Reserved
10011	AD19	Reserved	Reserved
10100	AD20	Reserved	Reserved
10101	AD21	Reserved	Reserved
10110	AD22	Reserved	Reserved
10111	AD23	Reserved	12-bit DAC0 Output/ ADC0_SE23
11000	AD24	Reserved	Reserved
11001	AD25	Reserved	Reserved
11010	AD26	Temperature Sensor (Diff)	Temperature Sensor (S.E)
11011	AD27	Bandgap (Diff) ²	Bandgap (S.E) ²
11100	AD28	Reserved	Reserved
11101	AD29	-VREFH (Diff)	VREFH (S.E)
11110	AD30	Reserved	VREFL
11111	AD31	Module Disabled	Module Disabled

Entradas Analógicas: Pinagem

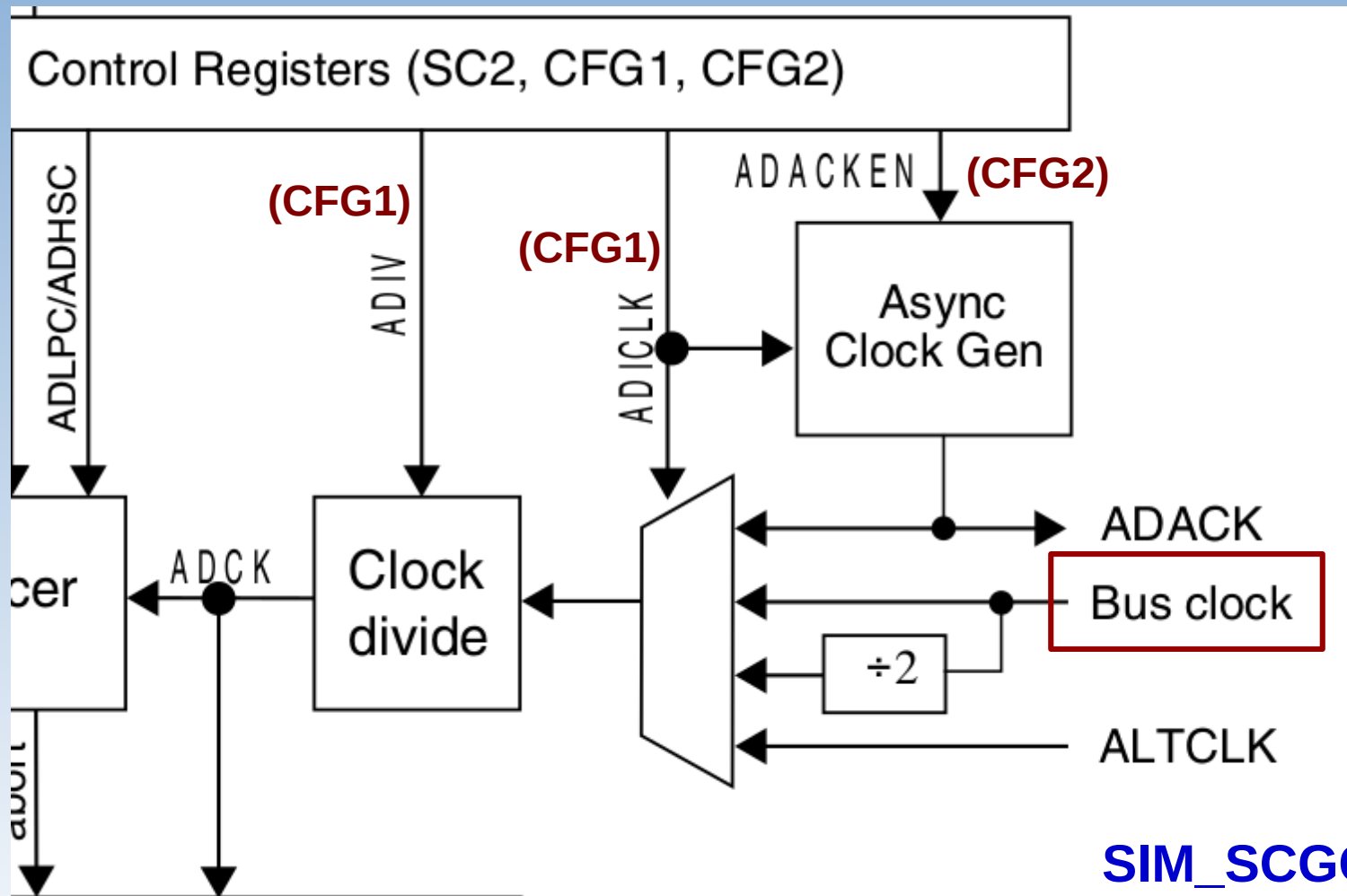
80 LQFP	64 LQFP	48 QFN	32 QFN	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
39	31	23	16	VSS	VSS	VSS							
40	32	24	17	PTA18	EXTAL0	EXTAL0	PTA18		UART1_RX	TPM_CLKIN0			
41	33	25	18	PTA19	XTAL0	XTAL0	PTA19		UART1_TX	TPM_CLKIN1		LPTMR0_ALT1	
42	34	26	19	RESET_b	RESET_b		PTA20						
43	35	27	20	PTB0/ ILWLL_P5	ADC0_SE8/ TSI0_CH0	ADC0_SE8/ TSI0_CH0	PTB0/ ILWLL_P5	I2C0_SCL	TPM1_CH0				
44	36	28	21	PTB1	ADC0_SE9/ TSI0_CH6	ADC0_SE9/ TSI0_CH6	PTB1	I2C0_SDA	TPM1_CH1				
45	37	29	—	PTB2	ADC0_SE12/ TSI0_CH7	ADC0_SE12/ TSI0_CH7	PTB2	I2C0_SCL	TPM2_CH0				
46	38	30	—	PTB3	ADC0_SE13/ TSI0_CH8	ADC0_SE13/ TSI0_CH8	PTB3	I2C0_SDA	TPM2_CH1				
47	—	—	—	PTB8	DISABLED		PTB8		EXTRG_IN				
48	—	—	—	PTB9	DISABLED		PTB9						
49	—	—	—	PTB10	DISABLED		PTB10	SPI1_PCS0					
50	—	—	—	PTB11	DISABLED		PTB11	SPI1_SCK					
51	39	31	—	PTB16	TSI0_CH9	TSI0_CH9	PTB16	SPI1_MOSI	UART0_RX	TPM_CLKIN0	SPI1_MISO		



Tensões de Referência

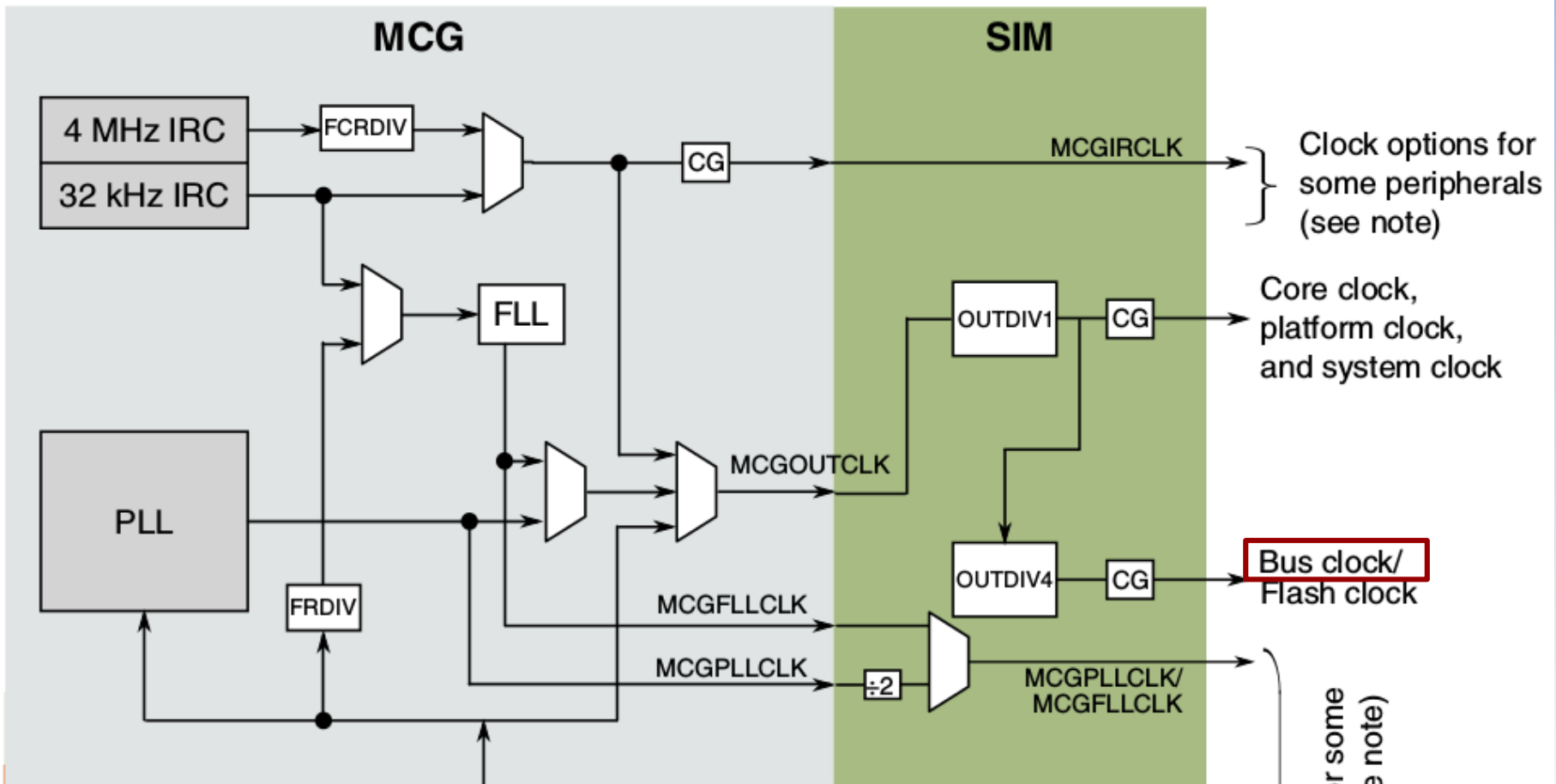


Fontes de Clock



Fontes de Clock

- MCGOUTCLK: 20,97152MHz

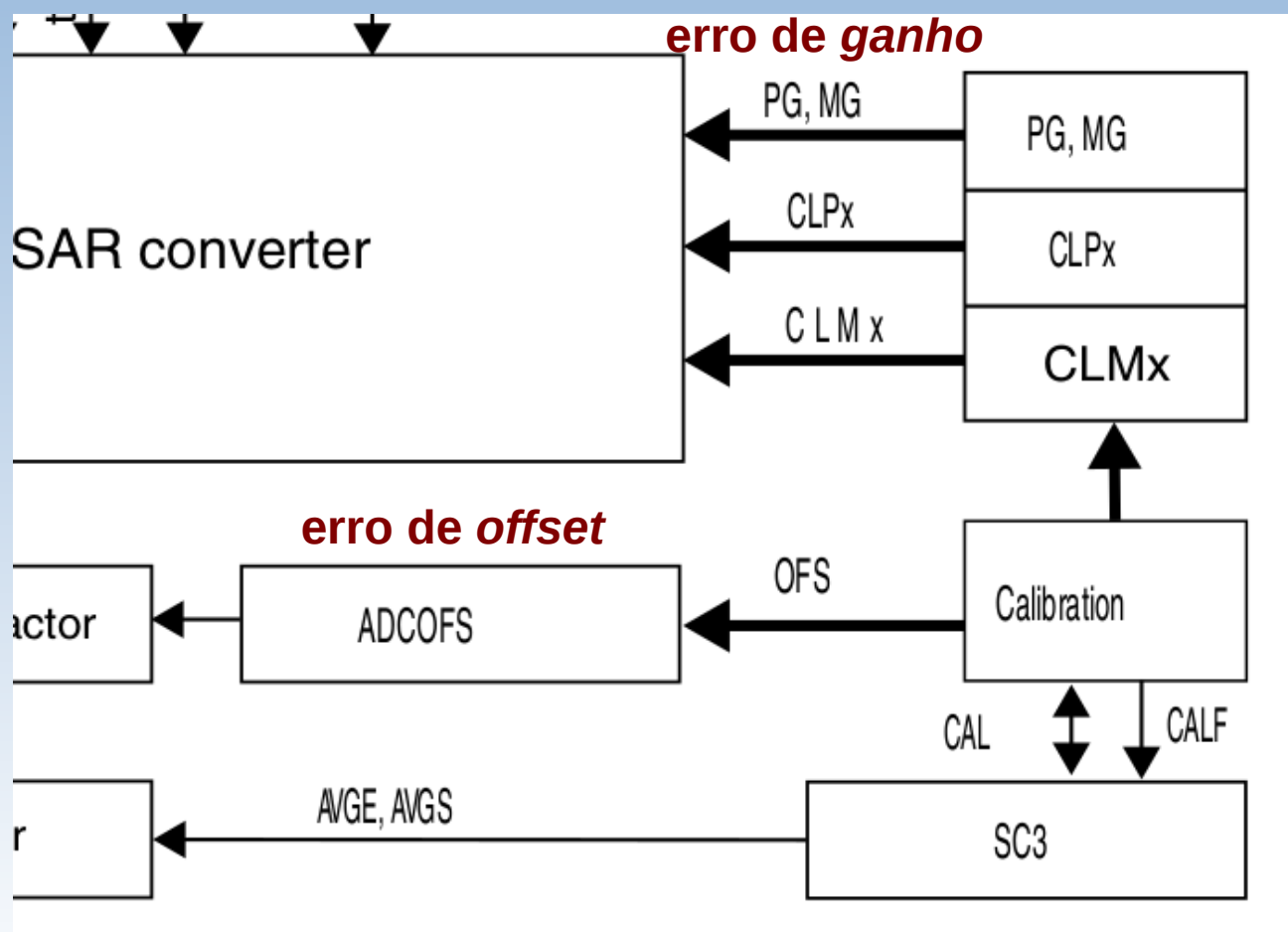


ADCK

Symbol	Description	Conditions	Min.	Typ. ¹	Max.	Unit	Notes
V_{ADIN}	Input voltage	<ul style="list-style-type: none"> 16-bit differential mode All other modes 	VREFL	—	31/32 * VREFH	V	—
C_{ADIN}	Input capacitance	<ul style="list-style-type: none"> 16-bit mode 8-bit / 10-bit / 12-bit modes 	—	8	10	pF	—
R_{ADIN}	Input series resistance		—	2	5	k Ω	—
R_{AS}	Analog source resistance (external)	13-bit / 12-bit modes $f_{ADCK} < 4$ MHz	—	—	5	k Ω	4
f_{ADCK}	ADC conversion clock frequency	\leq 13-bit mode	1.0	—	18.0	MHz	5
f_{ADCK}	ADC conversion clock frequency	16-bit mode	2.0	—	12.0	MHz	5
C_{rate}	ADC conversion rate	\leq 13-bit modes No ADC hardware averaging Continuous conversions enabled, subsequent conversion time	20.000	—	818.330	Ksps	6
C_{rate}	ADC conversion rate	16-bit mode No ADC hardware averaging Continuous conversions enabled, subsequent conversion time	37.037	—	461.467	Ksps	6

Calibração

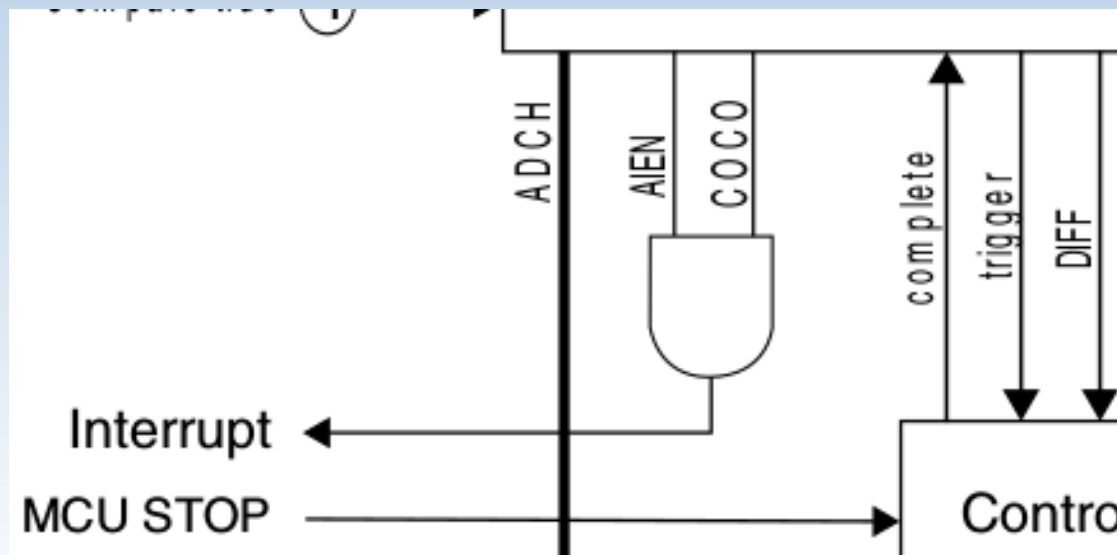
- Função de auto-calibração
- Compensações são automaticamente aplicadas na conversão.



Função de calibração: Seção 28.4.6 em *Reference Manual*

Compatibilização Temporal

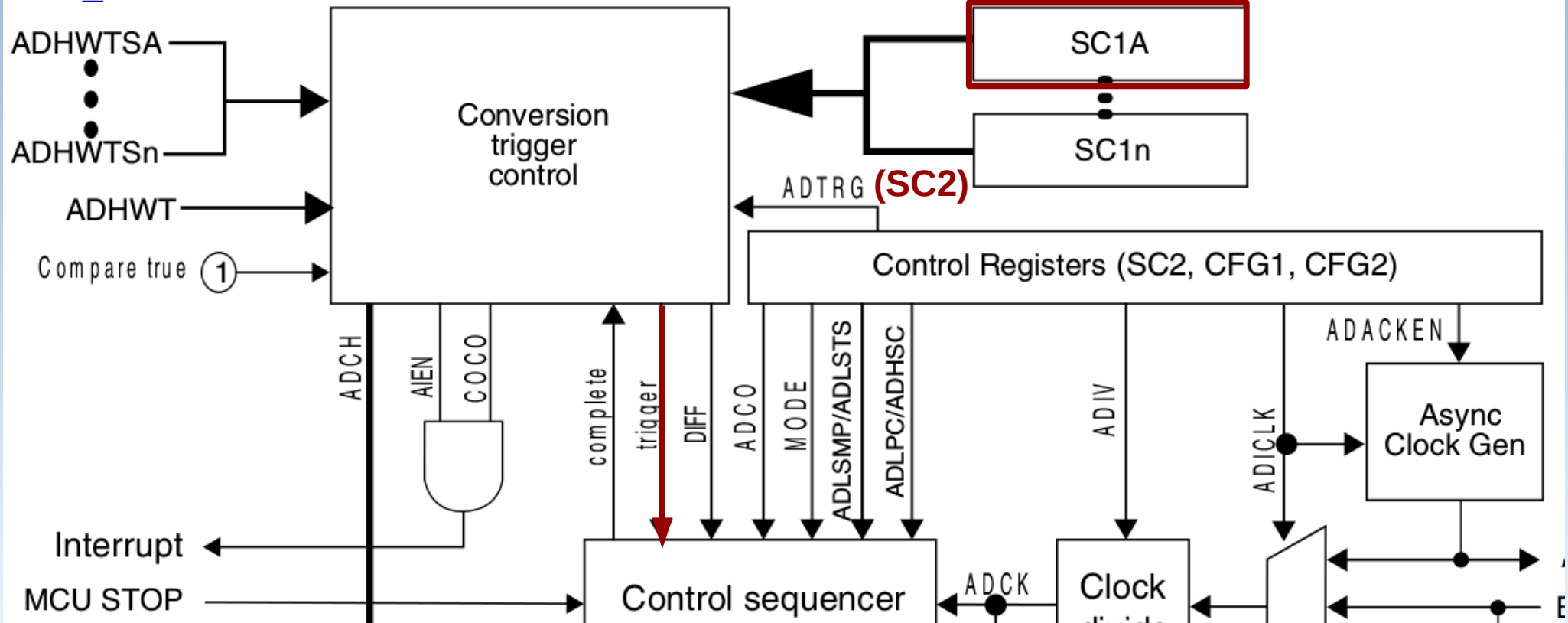
- Tempo de processamento na ordem de microsegundos
- *Flag* COCO: indicação do fim de uma conversão
 - Associado ao evento de interrupção
 - NVIC: IRQ15
 - Compatibilização temporal com o processador: *polling/* interrupção



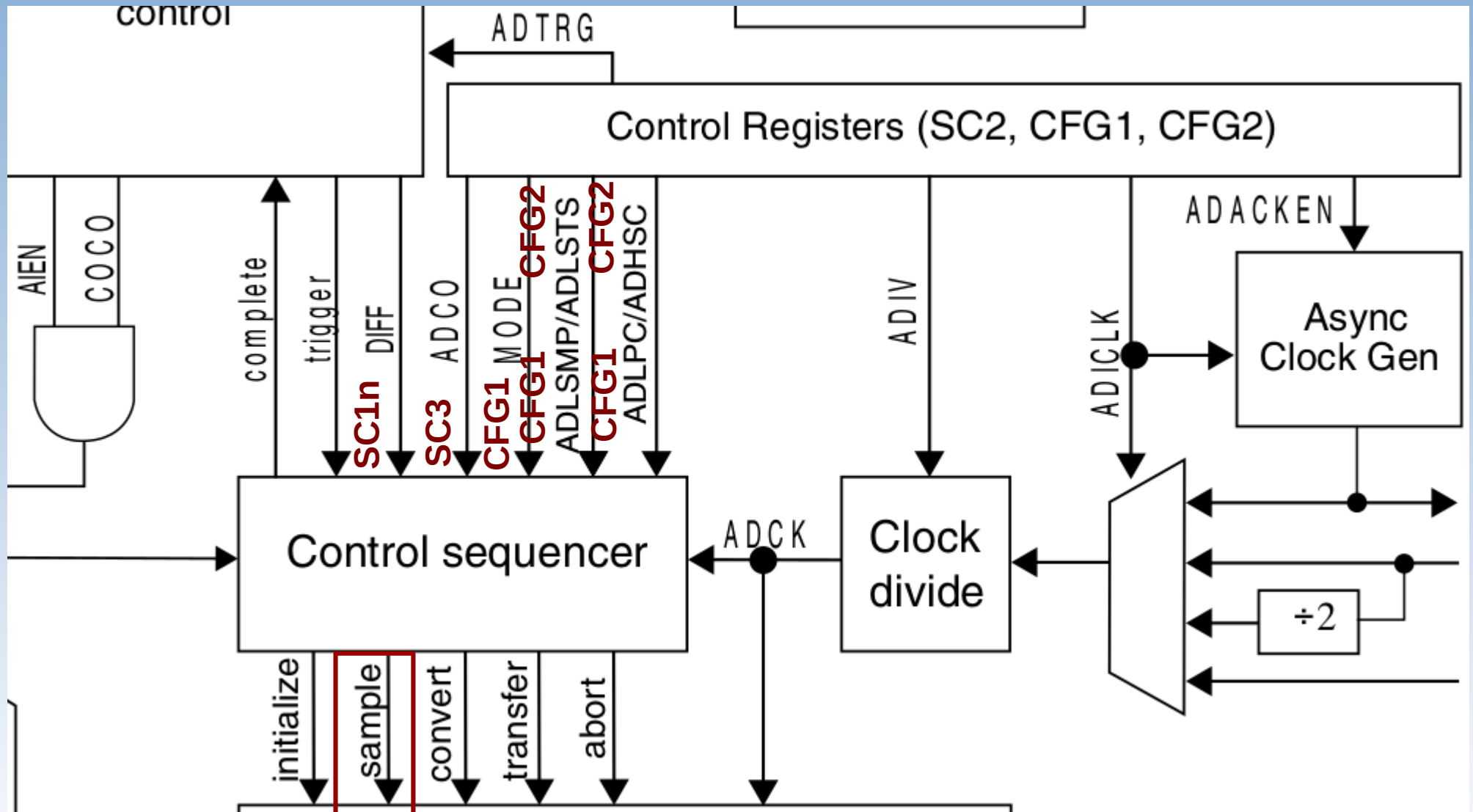
Triggers

(por hardware)

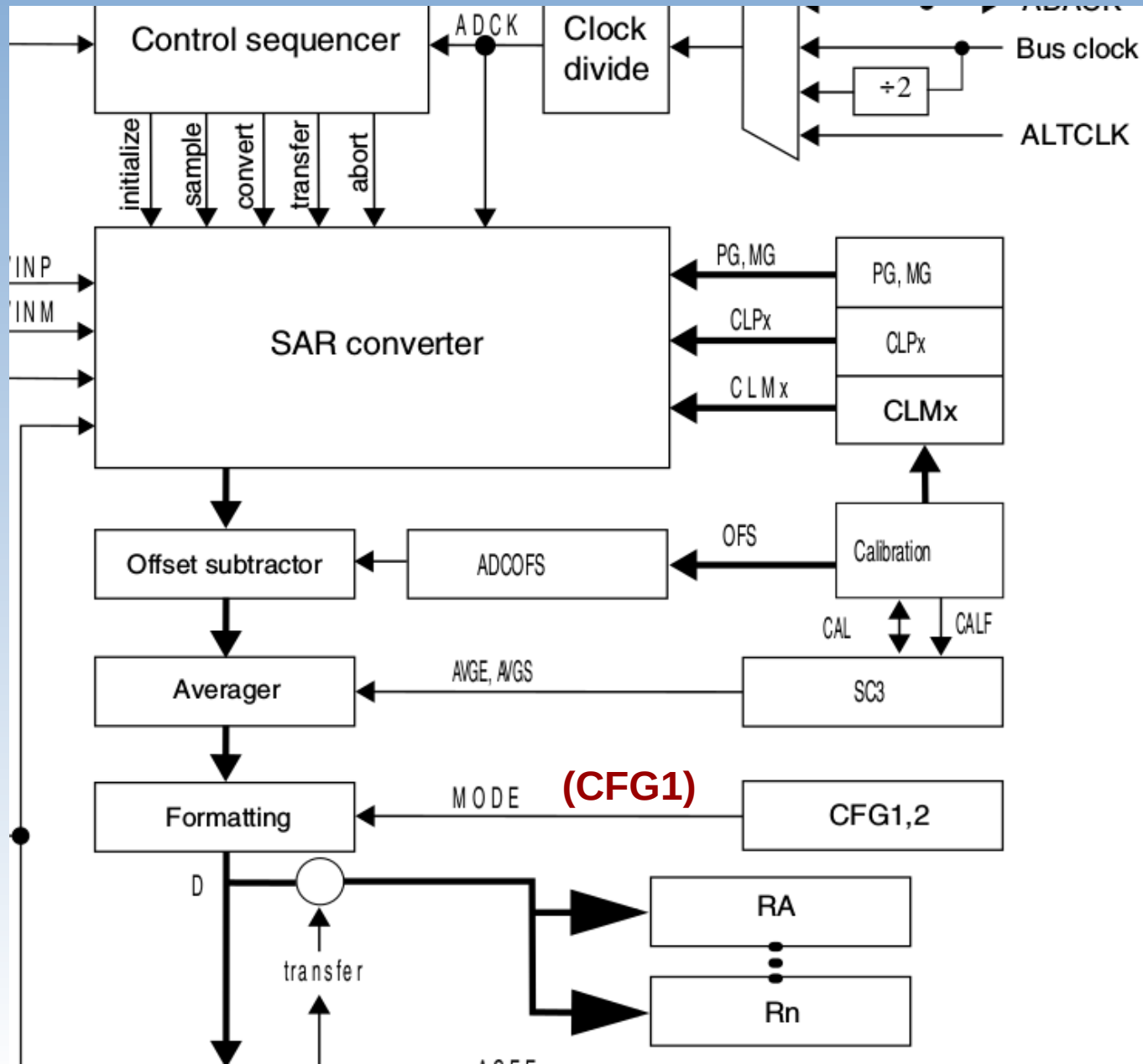
SIM_SOPT7



Controlador



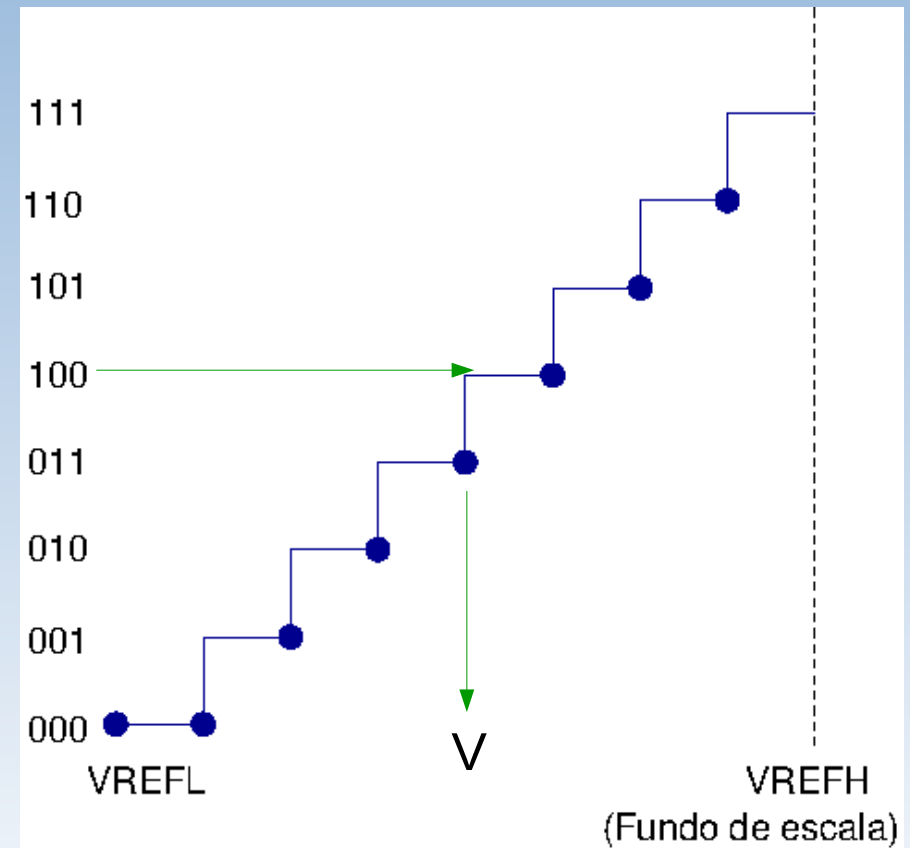
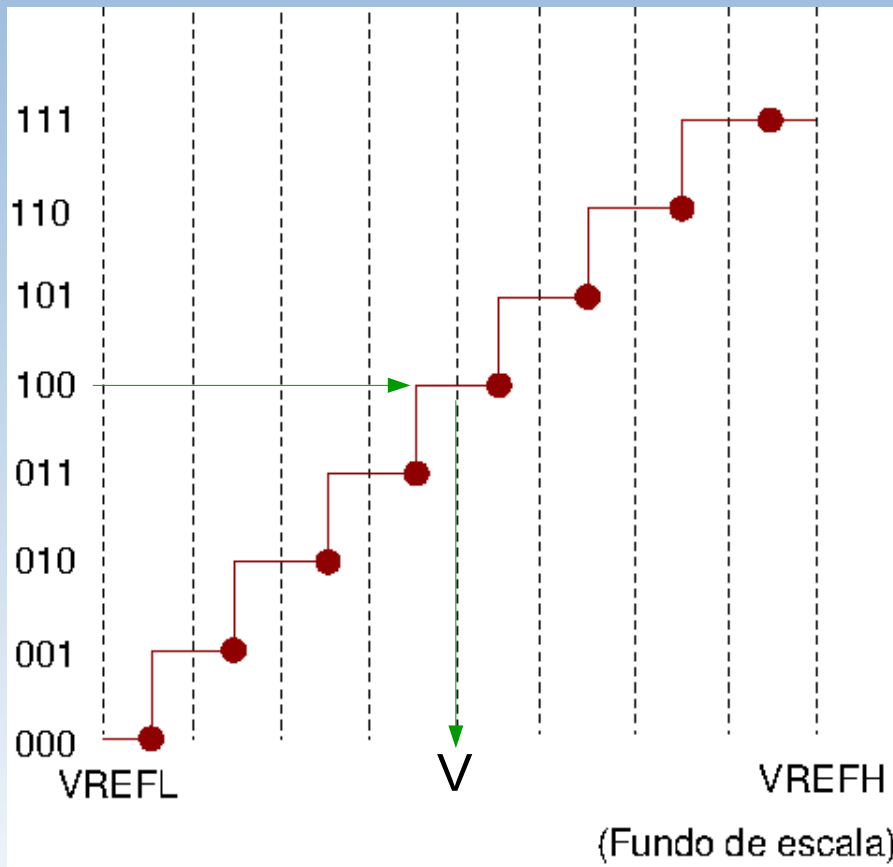
Conversão



Tensão x Código Binário

- 8, 10 e 12 *bits*: transições no meio do degrau

- 16 *bits*: transições no final do degrau



$$V = \frac{(VREFH - VREFL)}{(2^N)} * \text{Código Binário}$$

Tempo de Conversão

$$\text{ConversionTime} = \text{SFCAdder} + \text{AverageNum} \times (\text{BCT} + \text{LSTAdder} + \text{HSCAdder})$$

Figure 28-62. Conversion time equation

Table 28-70. Single or first continuous time adder (SFCAdder)

CFG1[AD LSMP]	CFG2[AD ACKEN]	CFG1[ADICLK]	Single or first continuous time adder (SFCAdder)
1	x	0x, 10	3 ADCK cycles + 5 bus clock cycles
1	1	11	3 ADCK cycles + 5 bus clock cycles ¹
1	0	11	5 μs + 3 ADCK cycles + 5 bus clock cycles
0	x	0x, 10	5 ADCK cycles + 5 bus clock cycles
0	1	11	5 ADCK cycles + 5 bus clock cycles ¹
0	0	11	5 μs + 5 ADCK cycles + 5 bus clock cycles

Tempo de Conversão

Table 28-71. Average number factor (AverageNum)

SC3[AVGE]	SC3[AVGS]	Average number factor (AverageNum)
0	xx	1
1	00	4
1	01	8
1	10	16
1	11	32

Table 28-72. Base conversion time (BCT)

Mode	Base conversion time (BCT)
8b single-ended	17 ADCK cycles
9b differential	27 ADCK cycles
10b single-ended	20 ADCK cycles
11b differential	30 ADCK cycles
12b single-ended	20 ADCK cycles
13b differential	30 ADCK cycles
16b single-ended	25 ADCK cycles
16b differential	34 ADCK cycles

Tempo de Conversão

Table 28-73. Long sample time adder (LSTAdder)

CFG1[ADLSMP]	CFG2[ADLSTS]	Long sample time adder (LSTAdder)
0	xx	0 ADCK cycles
1	00	20 ADCK cycles
1	01	12 ADCK cycles
1	10	6 ADCK cycles
1	11	2 ADCK cycles

Table 28-74. High-speed conversion time adder (HSCAdder)

CFG2[ADHSC]	High-speed conversion time adder (HSCAdder)
0	0 ADCK cycles
1	2 ADCK cycles

Tempo de Conversão

- Fonte de relógio: *bus clock* (20,97152MHz/1)
- Formato: 8 *bits* unipolar (mode = 0b00, diff=0)
- Divisor de frequência: *adiv* = 1 (*bus clock* = ADCK)
- Tempo de amostragem curto (*adlsmp* = 0)
- Conversão em alta velocidade habilitada (*adhsc*=1)

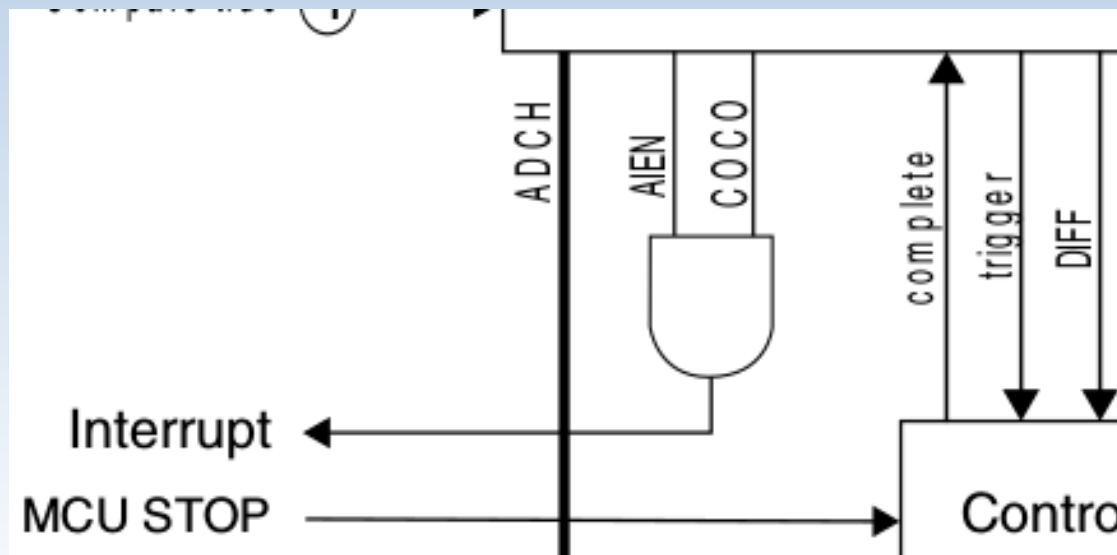
Table 28-77. Typical conversion time

Variable	Time
SFCAdder	5 ADCK cycles + 5 bus clock cycles
AverageNum	1
BCT	17 ADCK cycles
LSTAdder	0 ADCK cycles
HSCAdder	2

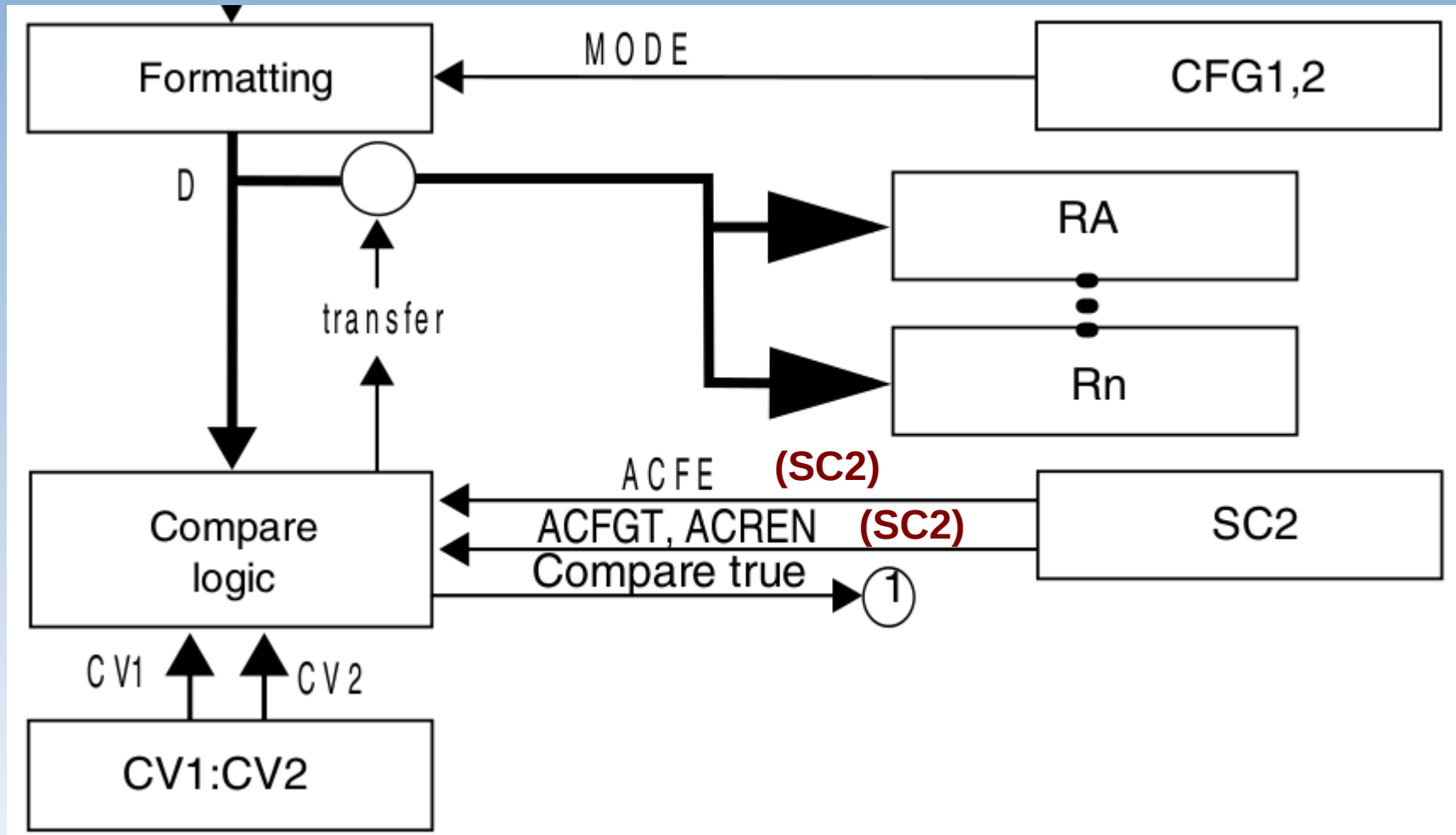
$$\textit{tempo de conversão} = \frac{(10 + 1 * (17 + 2))}{20971520} = 1,383 \mu s$$

Compatibilização Temporal

- Tempo de processamento na ordem de microsegundos
- *Flag* COCO: indicação do fim de uma conversão
 - Associado ao evento de interrupção
 - NVIC: IRQ15
 - Compatibilização temporal com o processador: *polling/* interrupção



Função de Comparação



Função de Comparação

Table 28-78. Compare modes

SC2[AC FGT]	SC2[AC REN]	ADCCV1 relative to ADCCV2	Function	Compare mode description
0	0	—	Less than threshold	Compare true if the result is less than the CV1 registers.
1	0	—	Greater than or equal to threshold	Compare true if the result is greater than or equal to CV1 registers.
0	1	Less than or equal	Outside range, not inclusive	Compare true if the result is less than CV1 Or the result is greater than CV2.
0	1	Greater than	Inside range, not inclusive	Compare true if the result is less than CV1 And the result is greater than CV2.
1	1	Less than or equal	Inside range, inclusive	Compare true if the result is greater than or equal to CV1 And the result is less than or equal to CV2.
1	1	Greater than	Outside range, inclusive	Compare true if the result is greater than or equal to CV1 Or the result is less than or equal to CV2.

Registadores

- SIM
 - SIM_SOPT4: *triggers por hardware*
 - SIM_SCGC6: habilitação da porta de clock
- ADCx
 - ADCx_SC1A: canal de entrada ativo, quantidade de sinais de entrada (unipolar/bipolar)
 - ADCx_CFG1: frequência de ADCK, tempo de amostragem
 - ADCx_CFG2: velocidade de conversão e tempo de amostragem
 - ADCx_SC2: tipo de *trigger*, função de comparação, tensões de referência
 - ADCx_SC3: fluxo de amostragem, média por *hardware*
 - ADCx_Rn: registrador de resultado de conversão
 - ADCx_Cvn : valores de comparação
 - Registradores de calibração
- PORTx
 - PORTx_PCRn: configuração da função do pino para sinal de entrada analógico
- NVIC
 - Gerenciamento da solicitação de interrupção IRQ 15

Programação

Inicialização ADC

Habilitar a porta de clock de módulo ADC.

Calibrar ADC em condições mais próximas possíveis das condições de operação de maior precisão (*polling*)

- máximo número de amostras para cômputo de média
- frequência de ADCK \leq 4MHz
- VREFH=VDDA.

Configurar o modo de operação do conversor com os registradores SC1A, CFG1, CFG2, SC2, SC3.

Configure os pinos de entrada com PORTx_PCRn.

Configurar NVIC para processar COCO por interrupção.

Se o *trigger* for por *software*, escrever o canal de entrada no registrador SC1A.

Se o *trigger* for por *hardware*, configurar a fonte de *trigger*.

Laço: aguardar por amostras para processamentos.

Projeto-Exemplo

- Projete um sistema que faz **continuamente** a amostragem dos sinais do sensor de temperatura AN3031 integrado no Kinetis (canal AD26) com os seguintes parâmetros de conversão:

Fonte de relógio: *bus clock* (20,97152MHz/1)

Formato: 8 *bits* unipolar (mode = 0b00, diff=0)

Divisor de frequência: *adiv* = 1 (*bus clock* = ADCK)

Tempo de amostragem curto (*adlsm* = 0)

Conversão em alta velocidade habilitada (*adhsc*=1)

- Tempo de conversão: 1,383us.

Técnicas de Programação

Uso do tipo de dados struct para facilitar a calibração e a configuração do módulo:

```
typedef struct _ADCCconfiguration {
    uint8_t sc1_diff;    // conversao unipolar (0) ou diferencial (1)
    uint8_t cfg1_adlpc;  // configuracao de baixa potencia
    uint8_t cfg1_ahdiv;  // divisor de frequencia do ADICLK
    uint8_t cfg1_adlsmp; // configuracao do tempo de amostragem
    uint8_t cfg1_mode;   // resolucao e formato da saida
    uint8_t cfg1_adiclk; // fonte de clock de entrada
    uint8_t cfg2_muxsel; // selecao do canal: A (0) e B (1)
    uint8_t cfg2_adacken; // habilitacao da saida de sinal de clock assincrono
    uint8_t cfg2_adhsc;   // habilitacao de operacao em alta velocidade
    uint8_t cfg2_adlsts;  // selecao de tempo de amostragem longo
    uint8_t sc2_adtrg;    // selecao de trigger: software (0) e hardware (1)
    uint8_t sc2_acfe;     // habilitacao de funcao de comparacao em igualdade
    uint8_t sc2_acfgt;    // habilitacao de funcao de comparacao em desigualdade
    uint8_t sc2_acren;    // habilitacao de comparacao de uma faixa
    uint8_t sc2_dmaen;    // habilitacao de DMA
    uint8_t sc2_refsel;   // selecao de tensoes de referencia
    uint8_t sc3_adco;     // habilitacao de amostragem continua (1) ou unica (0)
    uint8_t sc3_avge;     // habilitacao de media de amostras por hardware
    uint8_t sc3_avgs;     // selecao da quantidade de amostras por media
} ADCCConfig;
```

Técnicas de Programação

- Uso de variáveis globais para comunicação entre a rotina de serviço ADC0_IRQHandler() com o fluxo principal
 - cycle_flags
 - result0A

Técnicas de Programação

- Modularização de códigos

void ADC0_calibrate (ADCConfig *config): calibração

void ADC0_configure (ADCConfig *config): configuração

void ADC0_init (ADCConfig *config): inicialização

void ADC0_initSC1A (): escolha do *trigger* por *software*

void ADC0_enableInterrup (): habilita a interrupção

void ADC0_disableInterrup(): desabilita a interrupção

ADC0_selSingleChannel (uint8_t canal): seleção do canal de entrada a ser amostrado.

SIM_initBusClock (uint8_t OUTDIV4): configuração da frequência de *bus clock*

NVIC_enableADC0IRQ (char priority): configuração de NVIC para IRQ15

Pseudocódigo

```
SIM_initBusClock (0b000);
```

Seta os valores de parâmetros

```
ADC0_init(&config);
```

```
NVIC_enableADC0IRQ (1);
```

```
ADC0_enableInterrup ();
```

```
ADC0_selSingleChannel (0b11010);           // AD26
```

Laço de espera:



CodeWarrior IDE Development Suite

Informações Adicionais

- Tocci et al. Sistemas Digitais: Princípios e Aplicações.
- Bonnie Baker e Miro Oljaca. How the voltage reference affects ADC performance, Part 1, Texas Instruments Incorporated.
<http://www.ti.com/lit/pdf/SLYT331>
- Elnatan Chagas. Sistemas de Conversão AD e DA
<http://www.dsif.fee.unicamp.br/~elnatan/ee610/18a%20Aula.pdf>
- Kinetis KL25 Sub-Family Datasheet
<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/KL25P80M48SF0.pdf>

Informações Adicionais

- KL25 Sub-Family Reference Manual

<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/KL25P80M48SF0RM.pdf>

- NVIC: Capítulo 3 (páginas 51, 77)
- Clock Distribution: Capítulo 5 (página 123)
- SIM: Capítulo 12 (páginas 200, 207)
- ADC: Capítulo 28 (página 457)

- Kinetis L Peripheral Module Quick Reference (Rev. 0.09/2012)

<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/KLQRUG.pdf>



EA871

Conversores Analógico-Digital

Pós-processamento

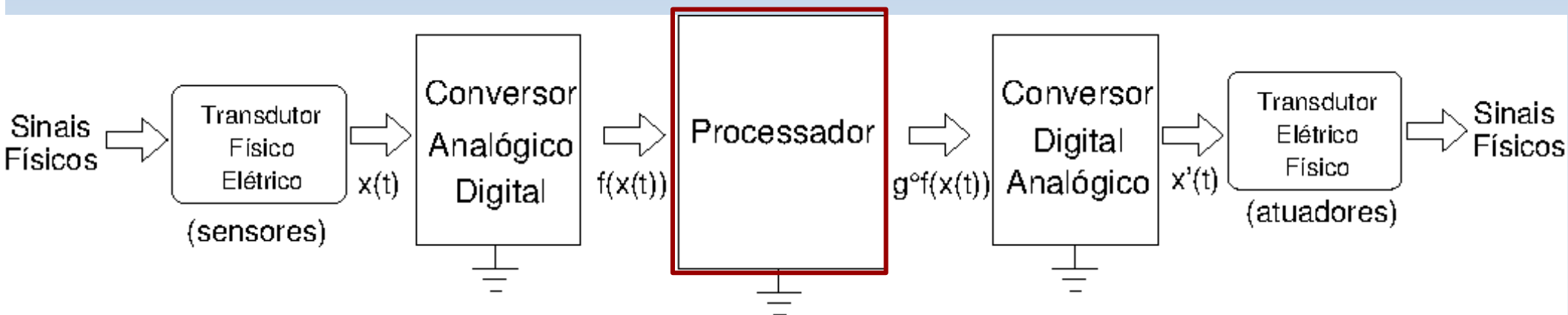
Wu Shin – Ting
DCA – FEEC - Unicamp
Segundo Semestre de 2020

Pós-Processamento

- Representação das amostras no formato esperado pelos algoritmos dedicados a uma tarefa específica – em grandezas físicas da aplicação
 - Código Binário → Tensões

$$(V - VREFL) = \frac{(VREFH - VREFL)}{(2^N - 1)} * \text{Código Binário}$$

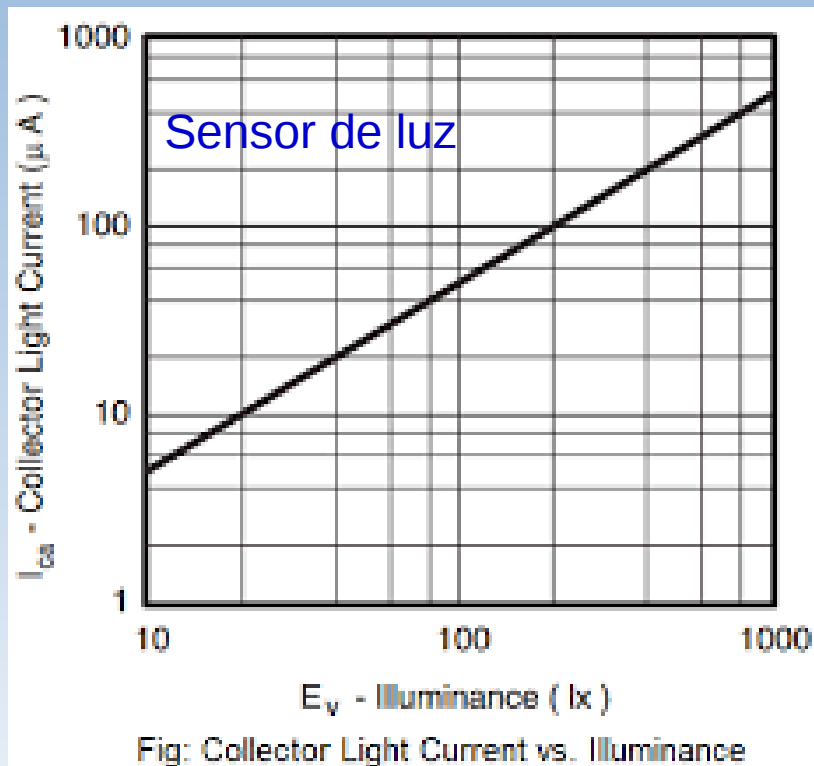
- **Tensões → Grandezas físicas**



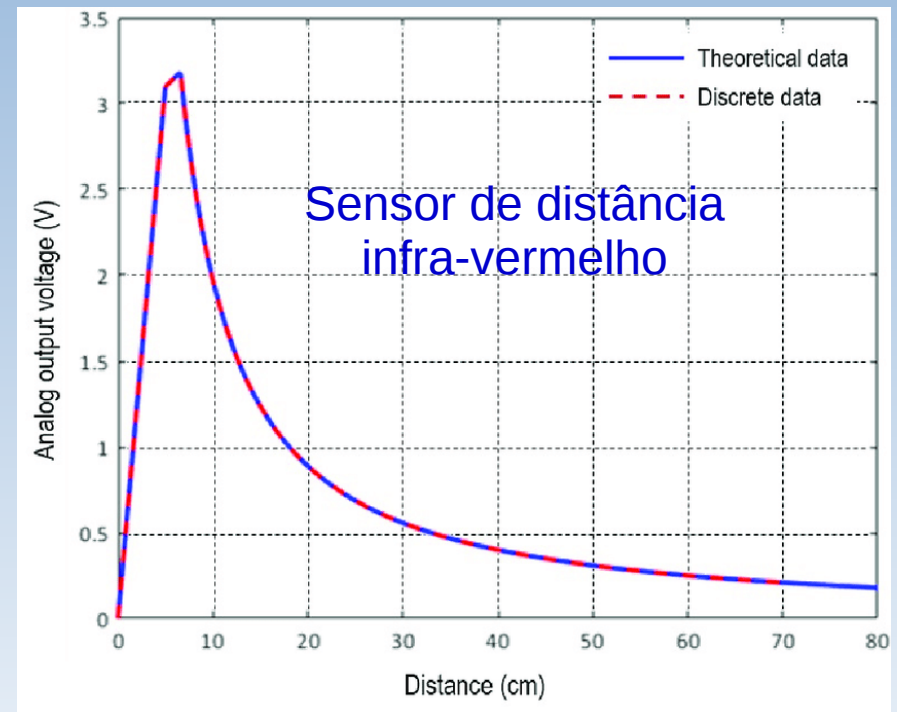
Grandezas físicas **Grandezas elétricas** **Código Binário**

Gráficos

- Relação entre uma grandeza física qualquer e uma grandeza elétrica.



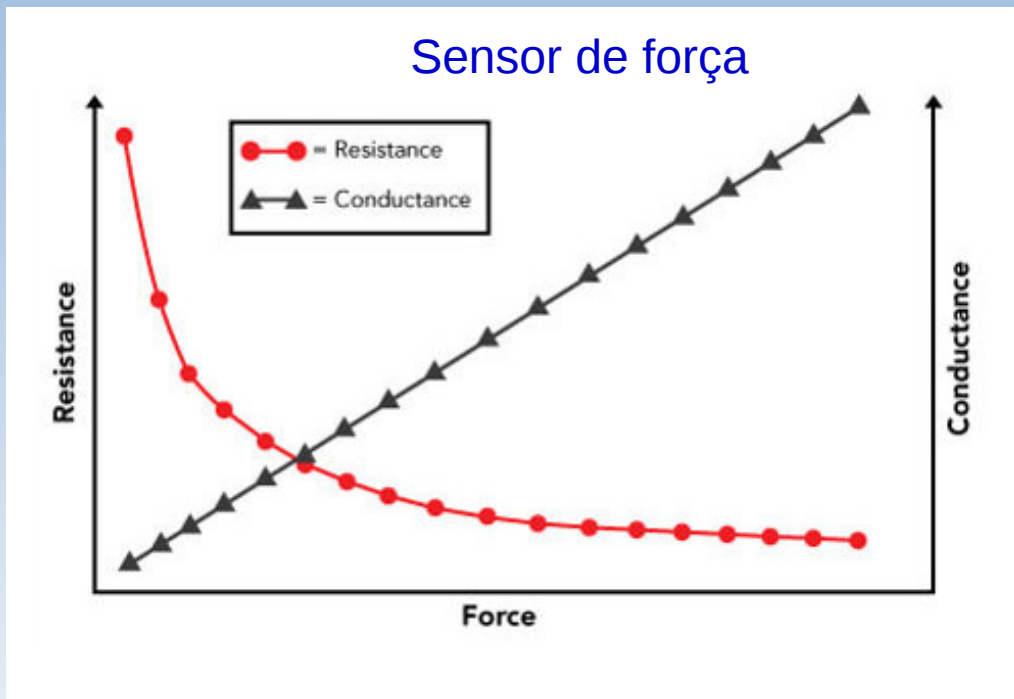
Iluminância x Corrente



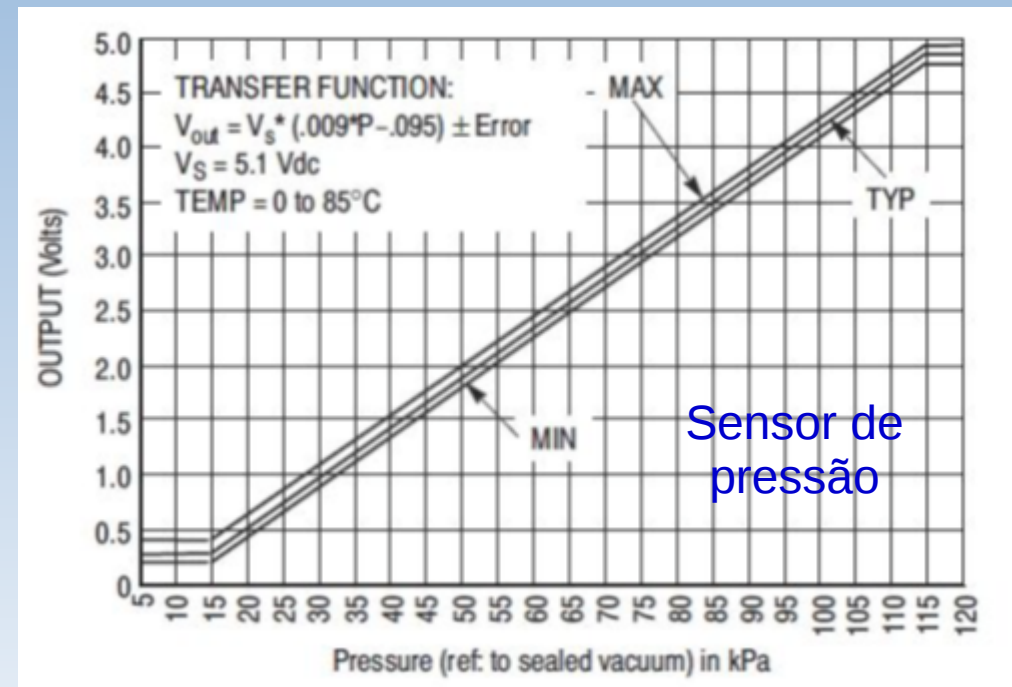
Distância x Tensão

Gráficos

- Relação entre uma grandeza física e uma grandeza elétrica.



Força x resistência

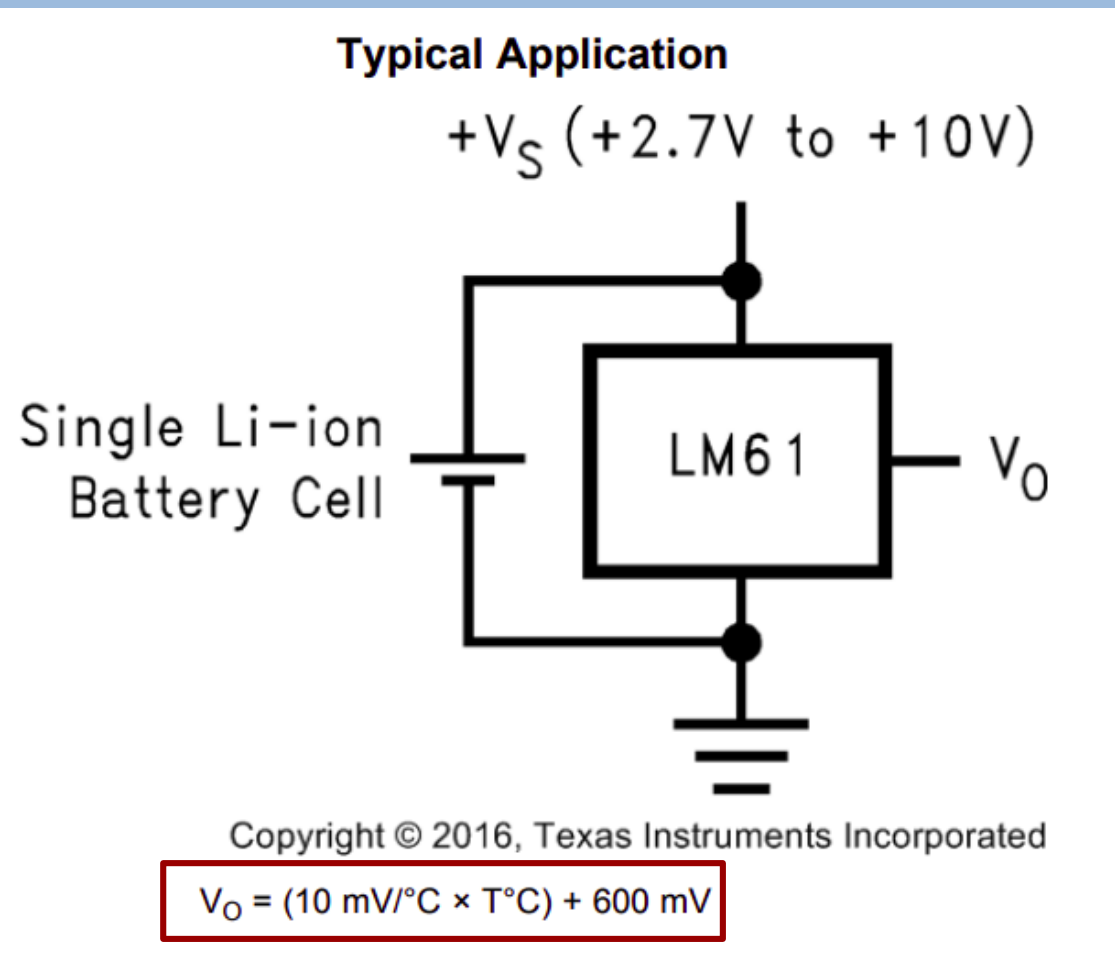


Pressão x Tensão

Fonte: *Datasheets na internet*

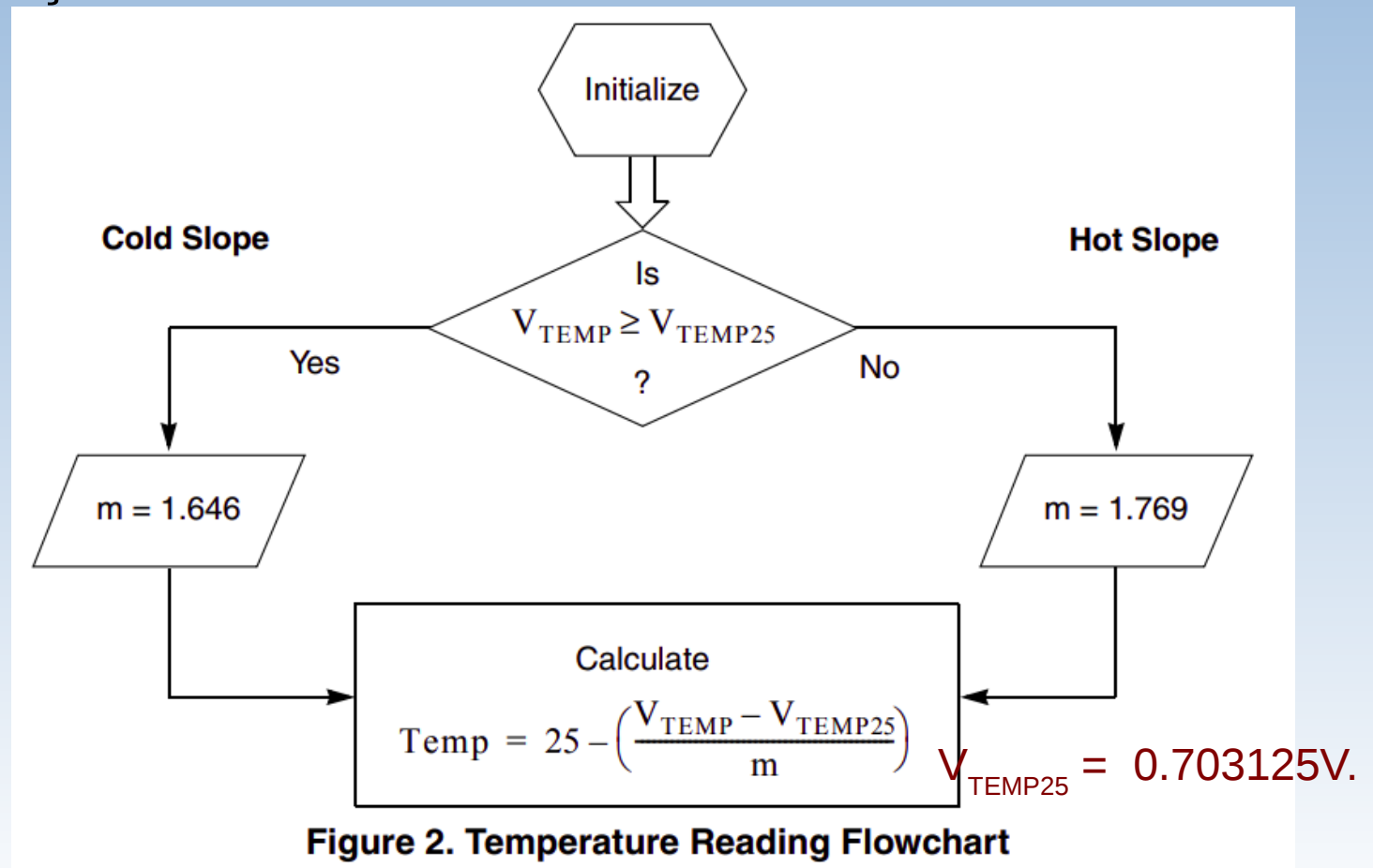
Sensor LM61

- Faixa de operação: -30°C a 100°C .
- Alimentação: 2.7V a 10.V



Sensor AN3031

- Sensor de temperatura para os microcontroladores da família HCS08.
- Faixa de operação: -40°C a 130°C



Projeto-Exemplo

- Projete um sistema que faz a amostragem dos sinais do sensor de temperatura AN3031 integrado no Kinetis (AD26) e de um sensor de temperatura LM61 (AD09) com os seguintes parâmetros de conversão:
 - Fonte de relógio: *bus clock* (20,97152MHz/1)
 - Formato: 10 *bits* unipolar (mode = 0b00, diff=0)
 - Divisor de frequência: *adiv* = 1 (*bus clock* = ADCK)
 - Tempo de amostragem curto (*adlsm* = 0)
 - Conversão em alta velocidade desabilitada (*adhsc*=0)
 - Tempo de conversão: 1,431us.
- Acesse os valores pós-processados na função main pela aba Variables no IDE CodeWarrior

PTB1: LM61

```
void PORTB_initTermometer(void) {  
    // Modulo SIM: habilita o clock do modulo PORTB  
    SIM_SCGC5 |= SIM_SCGC5_PORTB_MASK;  
  
    // Modulo PORT: configure o pino PTB1 para ADC0  
    PORTB_PCR1 &= ~PORT_PCR_MUX (0b111);  
}
```

Código Binários → Temperaturas

- Código Binário → Tensão ($V_{REFL}=0$; $V_{REFH}=3300mV$)

$$V = \frac{V_{REFH}}{(2^N - 1)} * \text{Código Binário}$$

- Tensão → Temperatura

- LM61
$$Temp = \frac{(V - 600 mV)}{10 \frac{mV}{C}}$$

- AN3031

$$Temp = 25 - \left(\frac{V - 703.125 mV}{m} \right)$$

$$m = 1,646, \text{ se } V < 703.125mV$$

$$m = 1,769, \text{ se } V \geq 703.125mV$$

Pseudocódigo

```
SIM_initBusClock (0b000);
```

Seta os valores de parâmetros;

```
ADC0_init(&config);
```

```
PORTB_initiTermometer();
```

Laço de espera:

```
    ADC0_selSingleChannel (0b11010);    //AD26
```

pós-processar a saída do conversor após a conclusão;

```
    ADC0_selSingleChannel (0b01001);    // AD09
```

pós-processar a saída do conversor após a conclusão;



CodeWarrior IDE Development Suite

Informações Adicionais

- *Datasheet* LM61

<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/datasheet/LM61.pdf>

- Temperature Sensot for the HCS08 Microcontroller Family

<https://www.nxp.com/docs/en/application-note/AN3031.pdf>



EA871

Conversores Analógico-Digital

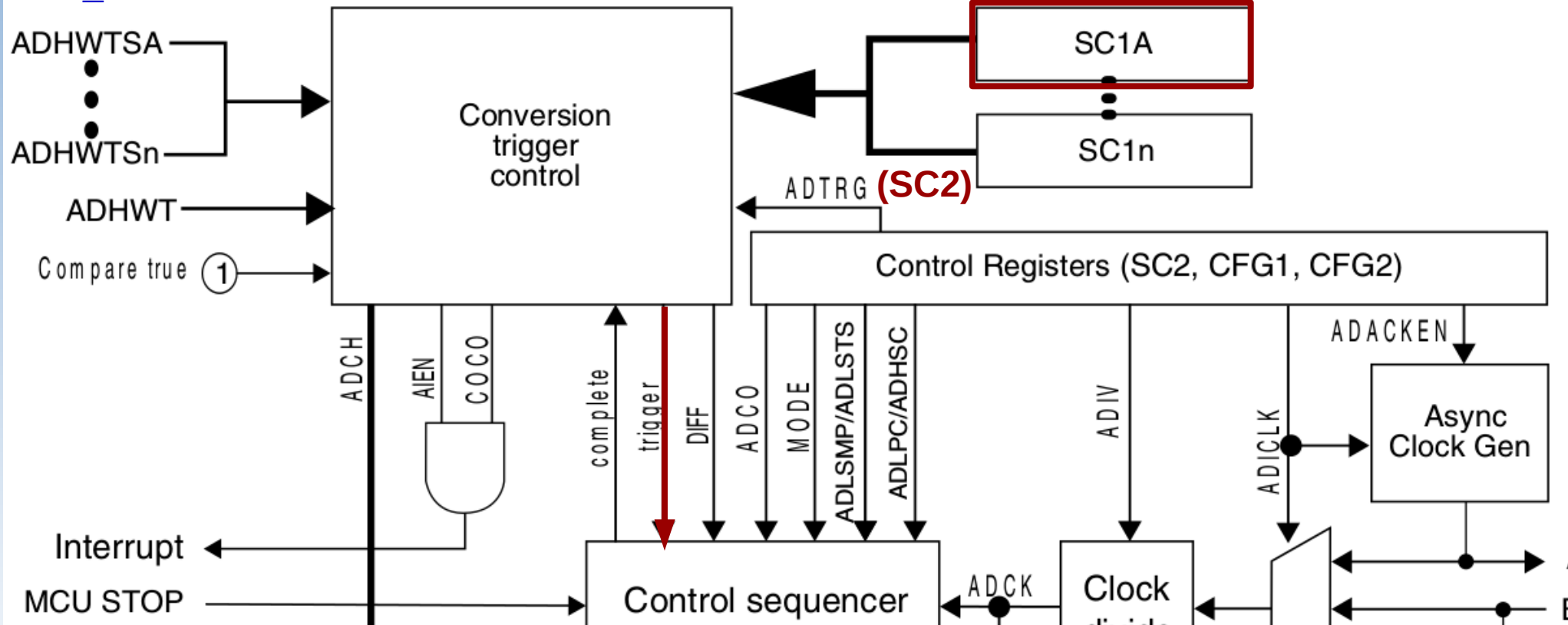
Conversão disparada por *hardware*

Wu Shin – Ting
DCA – FEEC - Unicamp
Segundo Semestre de 2020

Triggers

(por hardware)

SIM_SOPT7



Fontes de ADHWT

Table 3-1. Module to Module Interconnects

Peripheral	Signal	—	to Peripheral	Use Case	Control	Comment
TPM1	CH0F, CH1F	to	ADC (Trigger)	ADC Triggering (A AND B)	SOPT7_ADCAL TTRGEN = 0	Ch0 is A, and Ch1 is B, selecting this ADC trigger is for supporting A and B triggering. In Stop and VLPS modes, the second trigger must be set to >10us after the first trigger

Peripheral	Signal	—	to Peripheral	Use Case	Control	Comment
LPTMR	Hardware trigger	to	ADC (Trigger)	ADC Triggering (A or B)	SOPT7_ADC0T RGSEL (4 bit field), ADC0PRETRG SEL to select A or B	—
TPMx	TOF	to	ADC (Trigger)	ADC Triggering (A or B)	SOPT7_ADC0T RGSEL (4 bit field), SOPT7_ADC0P RETRGSEL to select A or B	—
PIT CHx	TIF0, TIF1	to	ADC (Trigger)	ADC Triggering (A or B)	SOPT7_ADC0T RGSEL (4 bit field), ADC0PRETRG SEL to select A or B	—
RTC	ALARM or SECONDS	to	ADC (Trigger)	ADC Triggering (A or B)	SOPT7_ADC0T RGSEL (4 bit field) ADC0PRETRG SEL to select A or B	—
EXTRG_IN	EXTRG_IN	to	ADC (Trigger)	ADC Triggering (A or B)	SOPT7_ADC0T RGSEL (4 bit field) ADC0PRETRG SEL to select A or B	—
CMP0	CMP0_OUT	to	ADC (Trigger)	ADC Triggering (A or B)	SOPT7_ADC0T RGSEL (4 bit field) ADC0PRETRG SEL to select A or B	—

<p style="text-align: center;">7 ADC0ALTTRGEN</p>	<p>ADC0 alternate trigger enable</p> <p>Enable alternative conversion triggers for ADC0.</p> <p>0 TPM1 channel 0 (A) and channel 1 (B) triggers selected for ADC0. 1 Alternate trigger selected for ADC0.</p>
<p style="text-align: center;">4 ADC0PRETRGSEL</p>	<p>ADC0 pretrigger select</p> <p>Selects the ADC0 pre-trigger source when alternative triggers are enabled through ADC0ALTTRGEN.</p> <p>0 Pre-trigger A 1 Pre-trigger B</p>
<p style="text-align: center;">3–0 ADC0TRGSEL</p>	<p>ADC0 trigger select</p> <p>Selects the ADC0 trigger source when alternative triggers are functional in stop and VLPS modes. .</p> <p>0000 External trigger pin input (EXTRG_IN) 0001 CMP0 output 0010 Reserved 0011 Reserved 0100 PIT trigger 0 0101 PIT trigger 1 0110 Reserved 0111 Reserved 1000 TPM0 overflow 1001 TPM1 overflow 1010 TPM2 overflow 1011 Reserved 1100 RTC alarm 1101 RTC seconds 1110 LPTMR0 trigger 1111 Reserved</p>

SIM_SOPT7

Projeto-Exemplo

- Baseado no projeto-exemplo apresentado no *Kinetis Quick Reference*:

Fonte de relógio: bus clock (20,97152MHz/1)

Formato: 16 bits unipolar (mode = 0b11, diff=0)

Divisor de frequência: adiv = 0b10 (bus clock/4 = ADCK)

Tempo de amostragem longo (adlsmpl = 1)

Conversão em alta velocidade habilitada (adhsc=1)

- *Trigger por hardware (adtrg=1)*
 - LPTMR substituído por PIT
- *Média de 32 amostras por hardware (avge=1, avgs = 0b11)*

Pseudocódigo

- 1) Turn on clocks to the ADC and **LPTMR PIT** module using the SIM module.
- 2) Configure System Integration Module for ADC trigger.
- 3) Configure the **LPTMR PIT**.
- 4) Determine the configuration the ADC using a structure to store the desired configuration.
- 5) Use the ADC driver to send the desired configuration to the ADC's.
- 6) Calibrate the ADCs in the configuration in which they will be used and then restore the desired configuration.
- 7) Enable the ADC and **LPTMR PIT** interrupts in NVIC.
- 8) Start **LPTMR PIT** counting and it will begin triggering ADC conversion periodically.
- 9) Handle the **LPTMR PIT** and ADC0 interrupts.

Detalhes de Implementação

- Passo 1:

```
SIM_SCGC5 |= SIM_SCGC5_LPTMR_MASK;
```

```
SIM_SCGC6 |= SIM_SCGC6_PIT_MASK;
```

- Passo 2:

```
SIM_SOPT7 |= (SIM_SOPT7_ADC0ALTTRGEN_MASK
```

```
 | !SIM_SOPT7_ADC0PRETRGSEL_MASK
```

```
 | SIM_SOPT7_ADC0TRGSEL(LPTMR0_TRG));
```

```
SIM_SOPT7 &= ~SIM_SOPT7_ADC0TRGSEL(0b1111);
```

```
SIM_SOPT7 |= (SIM_SOPT7_ADC0ALTTRGEN_MASK
```

```
 | SIM_SOPT7_ADC0TRGSEL(0b0100)) ;
```

```
SIM_SOPT7 &= ~SIM_SOPT7_ADC0PRETRGSEL_MASK;
```

```
ADC0_SC2 |= ADC_SC2_ADTRG_MASK;
```

Em C: Operador ! (NOT lógico) é diferente de ~(Complemento *bit a bit*).

Detalhes de Implementação

- Passo 3:

~~Configuração de LPTMR0~~

```
PIT_initTimer0(10485760); // Período de 0.5s
```

- Passo 4:

```
typedef struct _ADCConfiguration {
```

```
:
```

```
:
```

```
} ADCConfig
```

Detalhes de Implementação

- Passo 6:

~~Calibração ADC~~

```
ADC0_calibrate(ADCCConfig *config);
```

- Passo 7:

```
enable_irq(ADC0_irq_no);
```

```
enable_irq(LPTMR0_irq_no);
```

```
NVIC_enableADC0IRQ (1);
```

```
ADC0_enableInterrupt ();
```

Detalhes de Implementação

- Passo 8:

```
LPTMR0_CSR |= LPTMR_CSR_TEN_MASK;  
PIT_MCR &= ~(PIT_MCR_MDIS_MASK);
```

- Passo 9:

Interrupt service for ADC and LPTMR

```
void ADC0_IRQHandler(void) {  
    if( ADC0_SC1A & ADC_SC1_COCO_MASK ) {  
        result0A = ADC0_RA; // acesso de escrita limpa o flag COCO  
        cycle_flags = 1;  
    }  
}
```

- Passo 10:

- Ativar o canal de entrada



CodeWarrior IDE Development Suite

Informações Adicionais

- KL25 Sub-Family Reference Manual

<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/KL25P80M48SF0RM.pdf>

- NVIC: Capítulo 3 (páginas 51, 77)
 - Clock Distribution: Capítulo 5 (página 123)
 - SIM: Capítulo 12 (páginas 200, 207)
 - ADC: Capítulo 28 (página 457)
- Kinetis L Peripheral Module Quick Reference (Rev. 0.09/2012)

<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/KLQRUG.pdf>

- LPTMR + ADC: *Capítulo 11 (página 115)*