

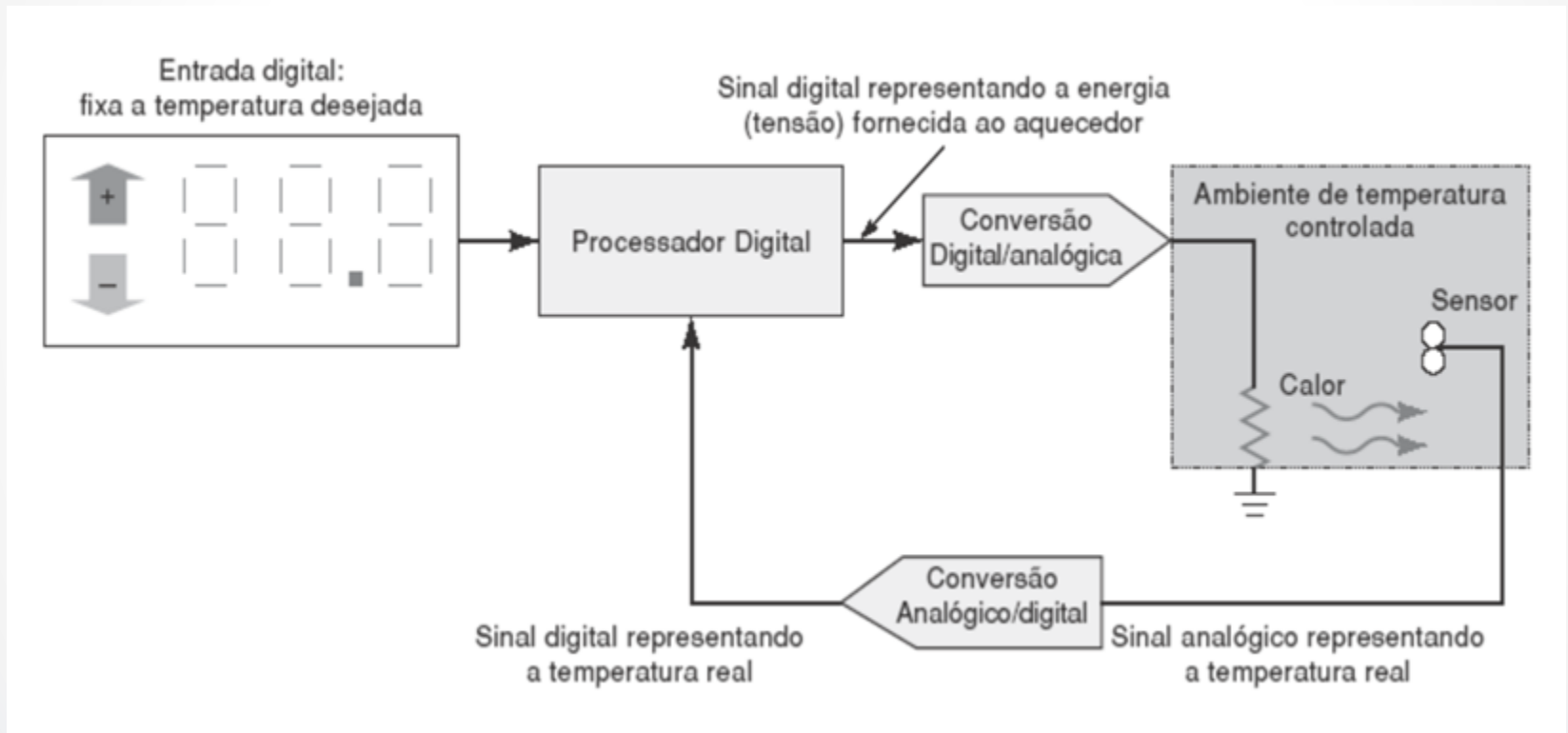
Sistemas Microcontrolados

Conversores D/A e A/D

Prof. Guilherme Peron

Introdução

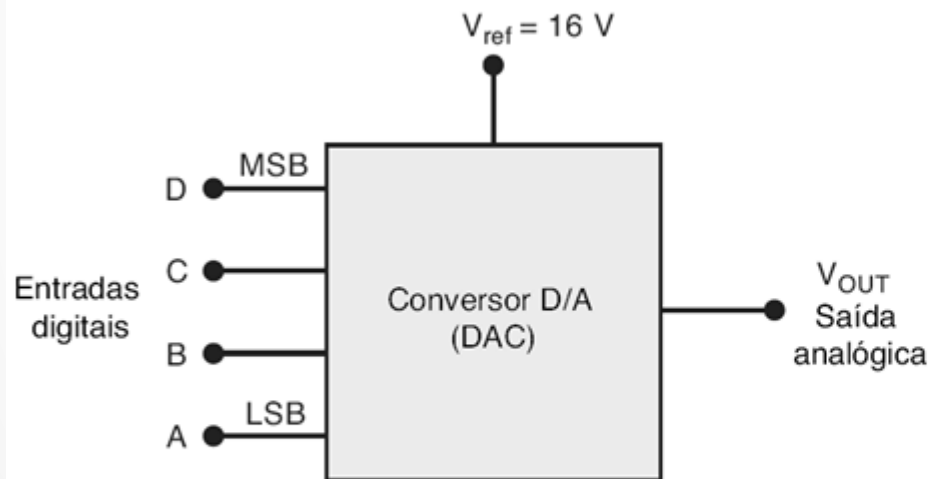
- Sistema Real



Conversor D/A

Conversor D/A

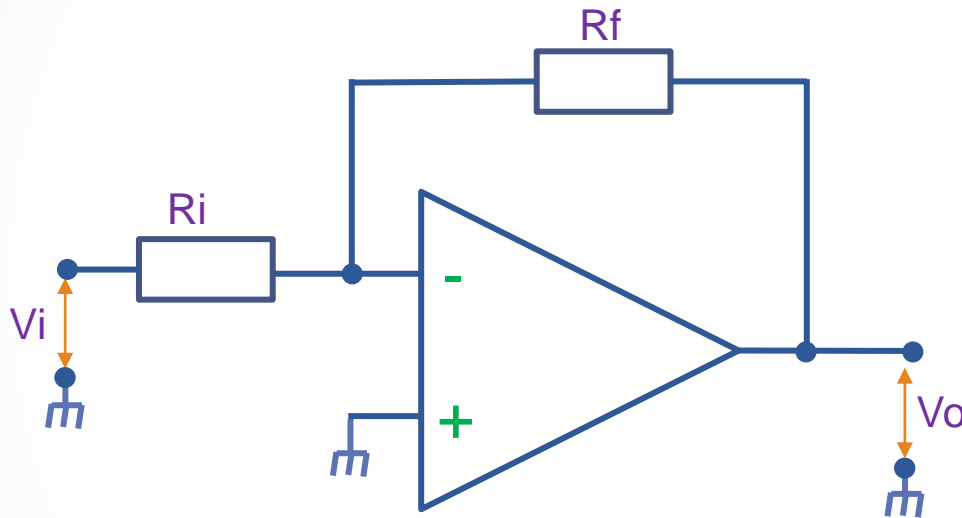
- A saída de um DAC não é tecnicamente uma quantidade analógica, porque pode assumir apenas valores específicos.



D	C	B	A	V_{OUT}	
0	0	0	0	0	Volts
0	0	0	1	1	↓
0	0	1	0	2	
0	0	1	1	3	
0	1	0	0	4	
0	1	0	1	5	
0	1	1	0	6	
0	1	1	1	7	
1	0	0	0	8	↓
1	0	0	1	9	
1	0	1	0	10	
1	0	1	1	11	
1	1	0	0	12	
1	1	0	1	13	
1	1	1	0	14	
1	1	1	1	15	
					Volts

Conversor D/A

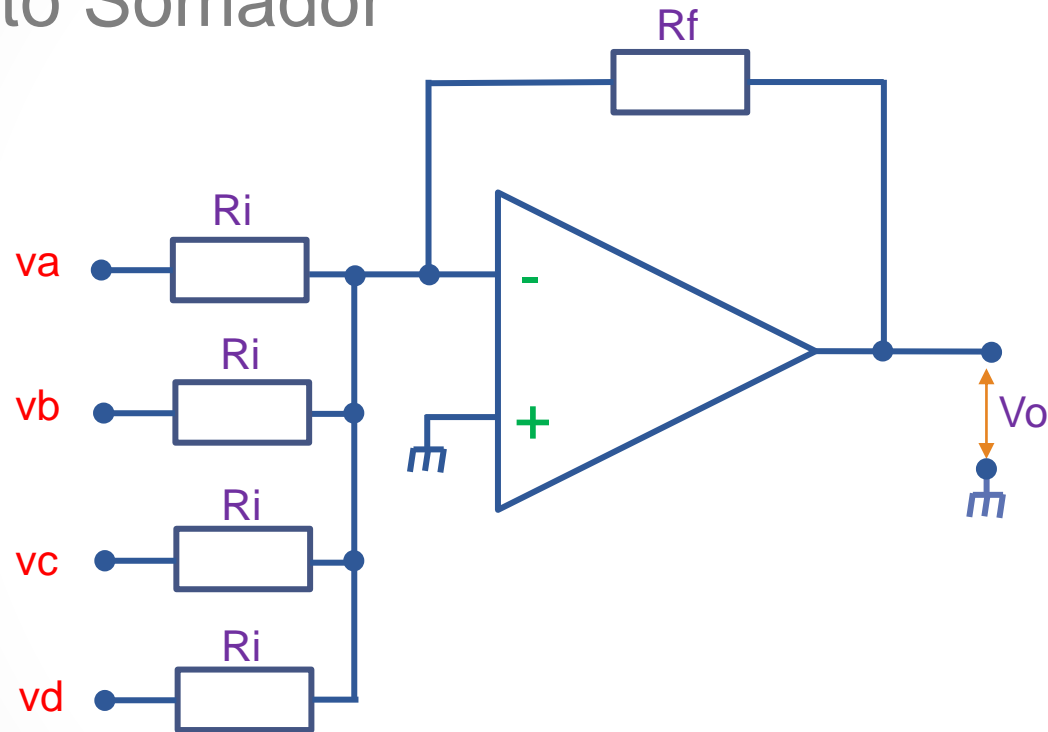
- Amplificador Inversor



$$V_o = -V_i \times (R_f / R_i)$$

Conversor D/A

- Circuito Somador



$$V_o = - (R_f / R_i) \times (v_a + v_b + v_c + v_d)$$

Conversor D/A

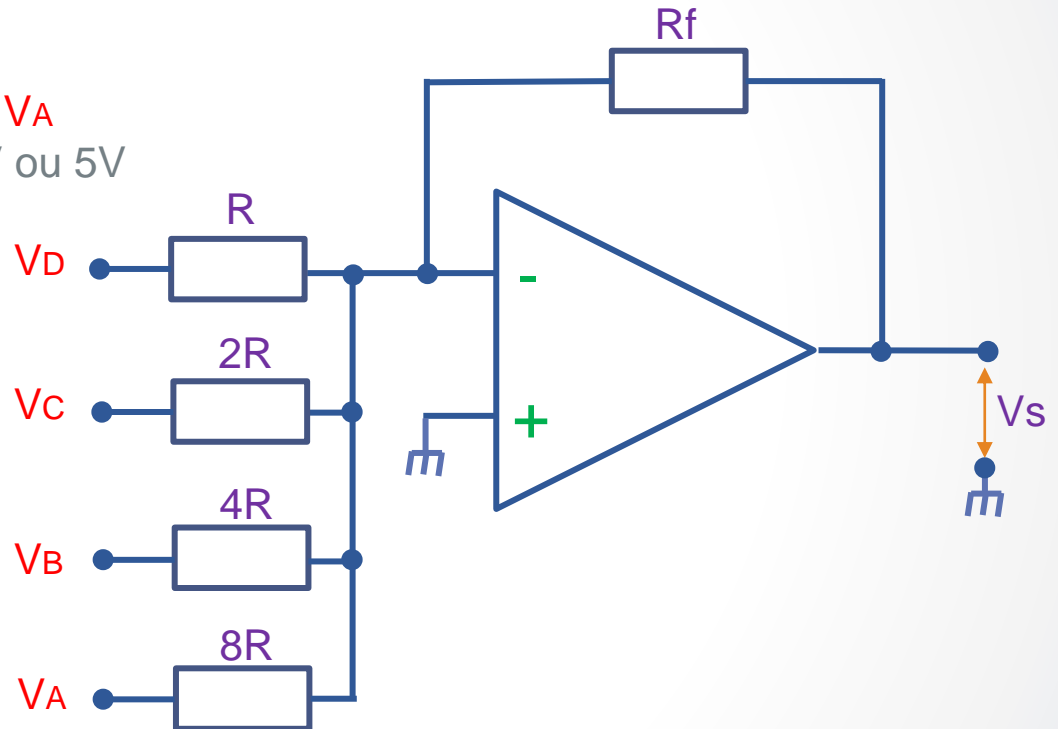
- Conversor D/A com Resistores Ponderados

Dados				Vs
D	C	B	A	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Exemplo

V_D, V_C, V_B, V_A

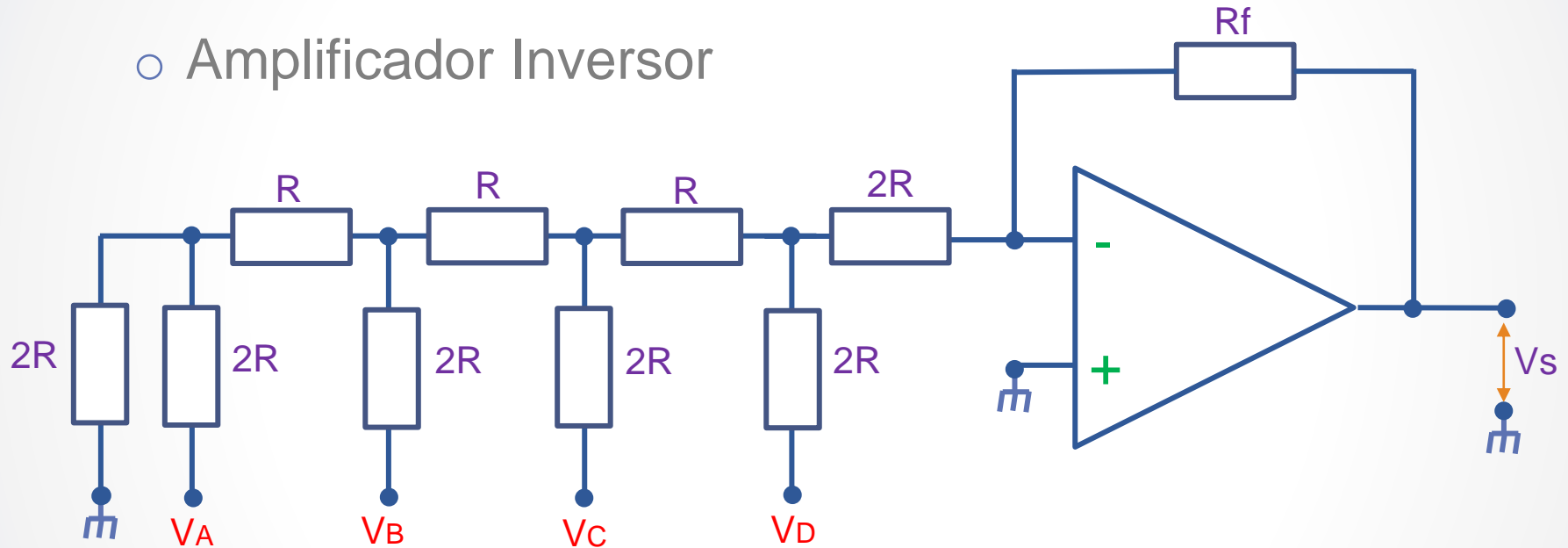
"1" → 3,3V ou 5V



$$V_s = - (R_f / R_i) \times (V_D + V_C/2 + V_B/4 + V_A/8)$$

Conversor D/A

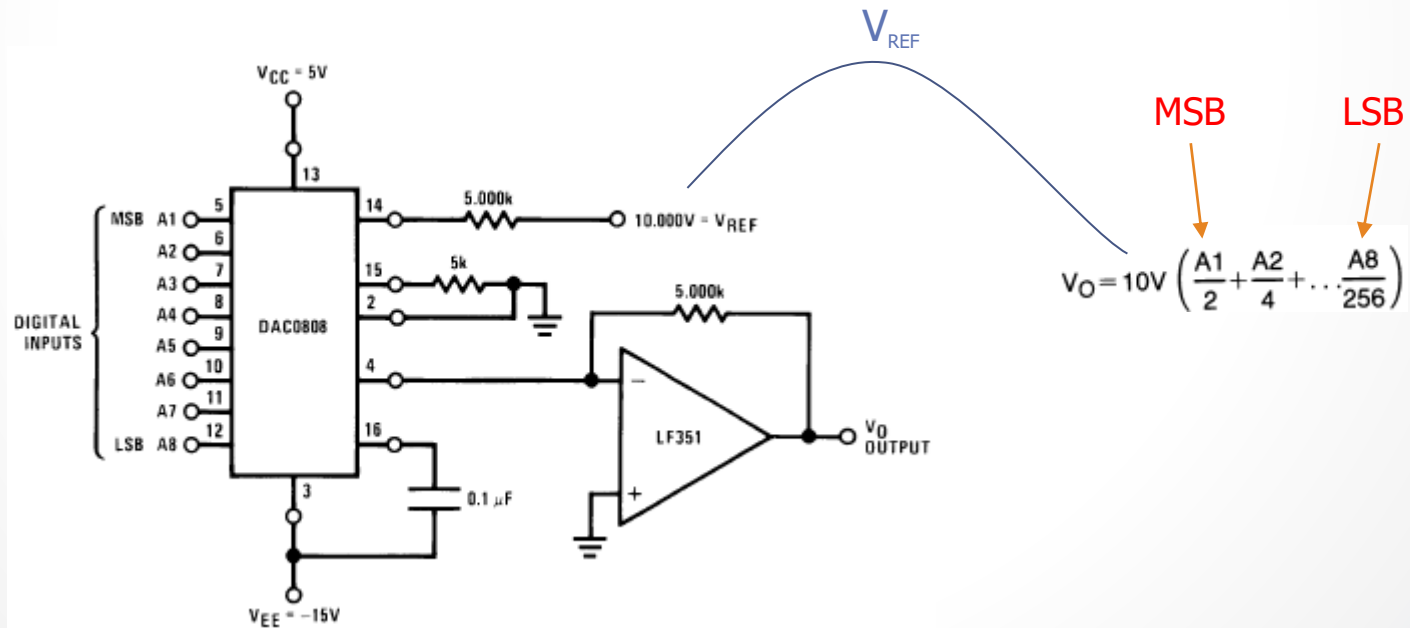
- Conversor D/A com Rede R/2R
 - Amplificador Inversor



$$V_s = - (R_f / 6R) \times (V_D + V_C/2 + V_B/4 + V_A/8)$$

Conversor D/A

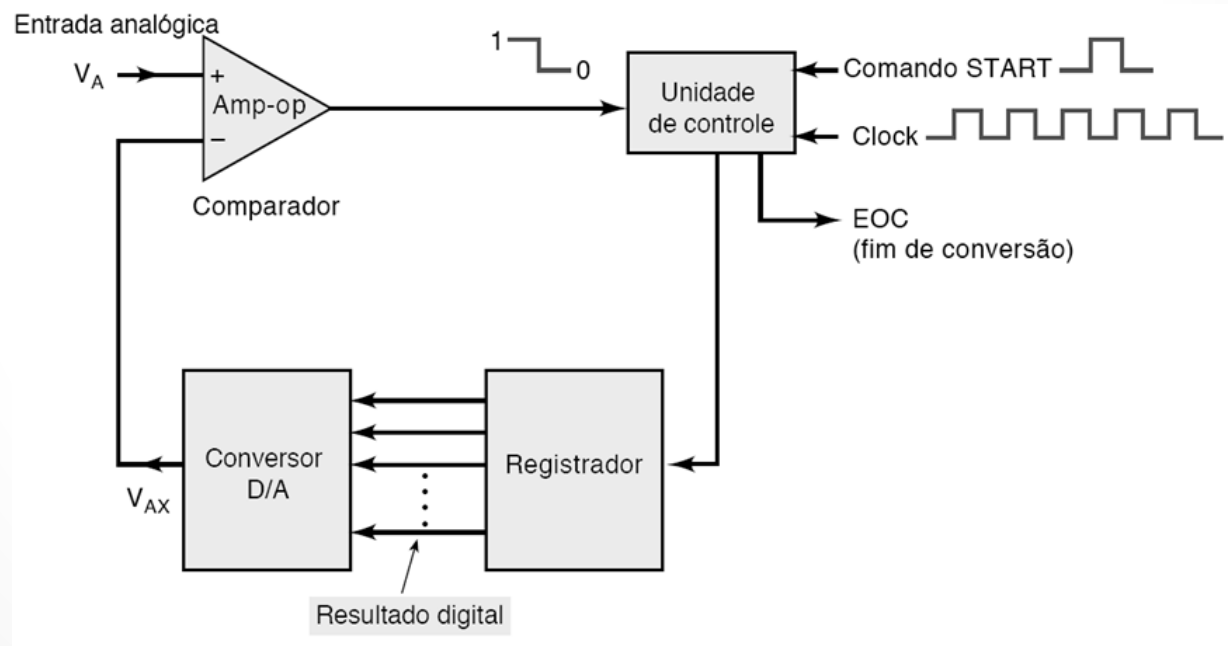
- Conversores Comerciais
 - Conversor D/A DAC0808
 - 8 bits de resolução, interface paralela
 - Tempo de acomodação: 150ns



Conversor A/D

Conversor A/D

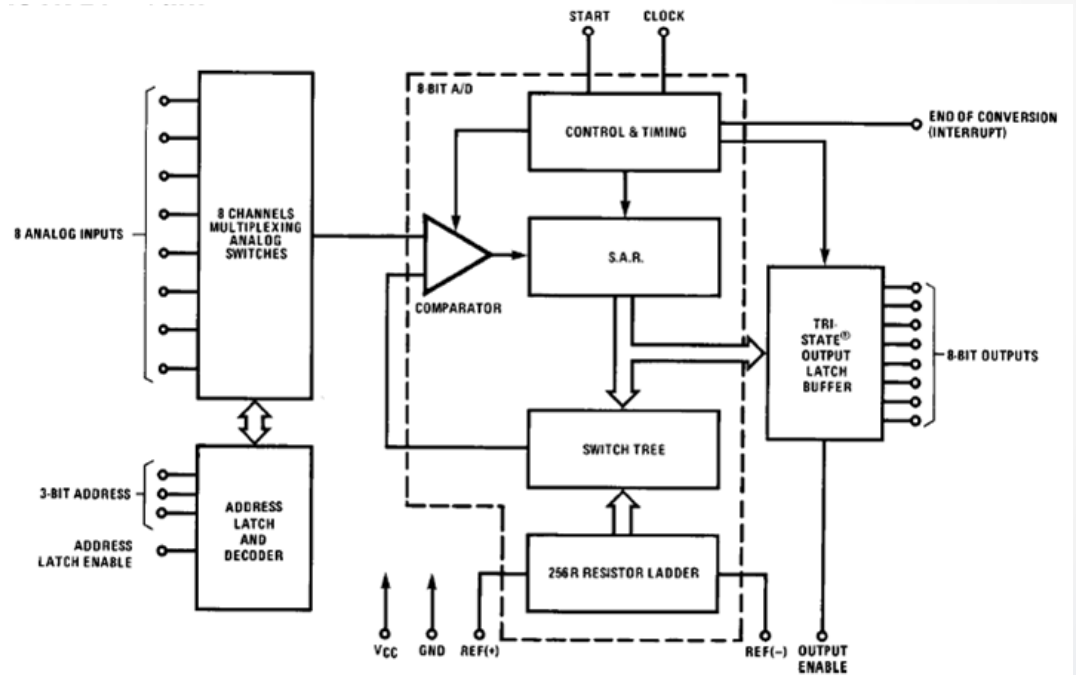
- Recebe uma tensão de entrada analógica e produz um código de saída digital que representa a entrada analógica.



Conversor A/D

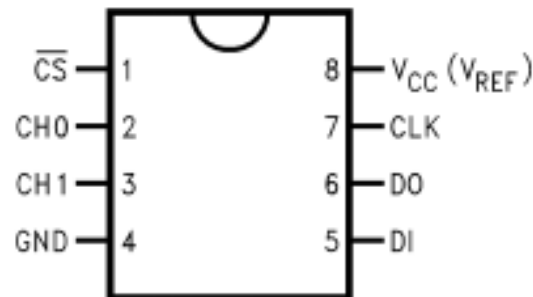
- ADC0808
 - 8 bits, 8 canais mux, $V_{in}=0-5V$

SELECTED ANALOG CHANNEL	ADDRESS LINE		
	C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H



Conversor A/D

- ADC0832
 - Interface SPI



Conversor A/D na Tiva

Características

- Possui 20 canais compartilhados para a conversão A/D (Tabela 15-1)

Pin Name	Pin Number	Pin Mux / Pin Assignment	Pin Type	Buffer Type	Description
AIN0	12	PE3	I	Analog	Analog-to-digital converter input 0.
AIN1	13	PE2	I	Analog	Analog-to-digital converter input 1.
AIN2	14	PE1	I	Analog	Analog-to-digital converter input 2.
AIN3	15	PE0	I	Analog	Analog-to-digital converter input 3.
AIN4	128	PD7	I	Analog	Analog-to-digital converter input 4.
AIN5	127	PD6	I	Analog	Analog-to-digital converter input 5.
AIN6	126	PD5	I	Analog	Analog-to-digital converter input 6.
AIN7	125	PD4	I	Analog	Analog-to-digital converter input 7.

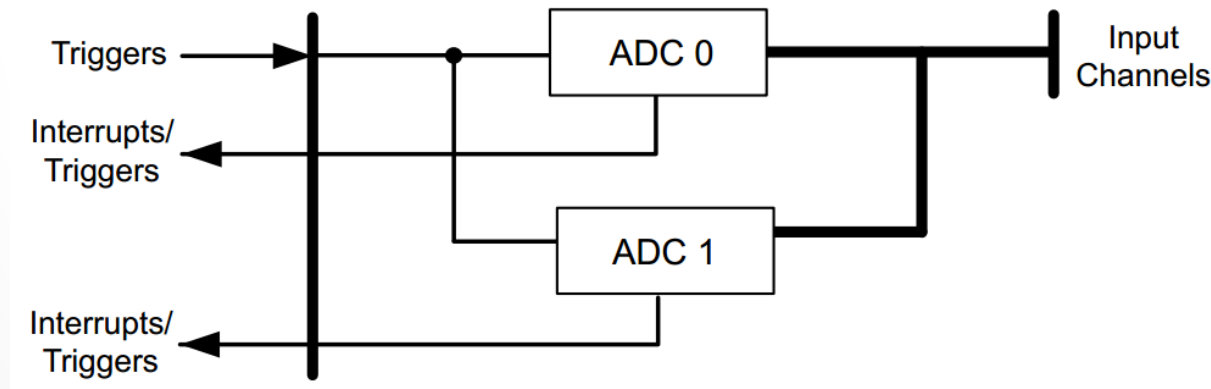
...

- Cada canal tem precisão de 12 bits
- Resolução

$$Res = \frac{\Delta V}{2^N} = \frac{3,3 - 0}{4096} = 805,66\mu V$$

Características

- Há dois módulos idênticos (ADC0 e ADC1)
- Ambos compartilham os 20 canais
- Cada ADC pode amostrar independentemente qualquer um dos canais e gerar interrupções



Características

- Há 4 sequenciadores de conversão das amostras (SS) com a possibilidade de 1 até 8 amostras
 - SS0 → 8 amostras
 - SS1 → 4 amostras
 - SS2 → 2 amostras
 - SS3 → 1 amostra

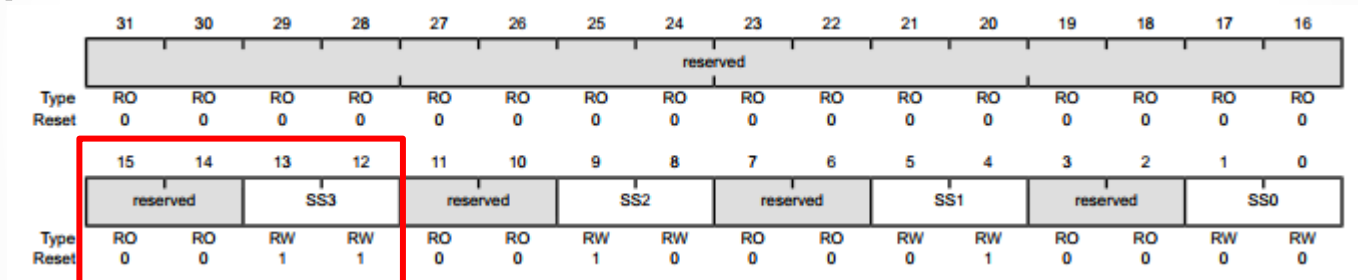


Registradores

- Para ativar o ADC, o *clock* do respectivo ADC tem que ser ativado no registrador **RCGCADC**
- Verificar o bit do ADC respectiva no registrador **PRADC** para saber se está pronto para o uso.

Registadores

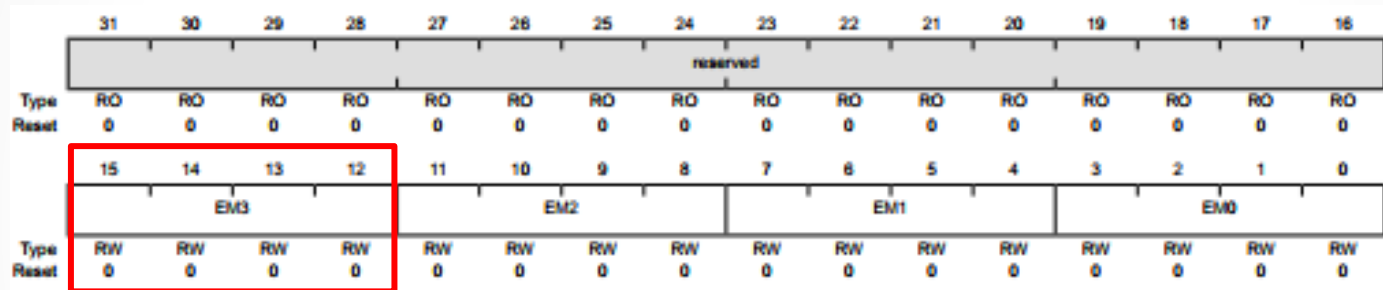
- O registrador **ADCSSPRI** contém as prioridades dos sequenciadores. Os dois bits inferiores de cada *nibble* guarda a prioridade de cada um dos sequenciadores.



- Cada sequenciador deve ter uma prioridade diferente do outro.
- Para aplicações simples: utilizar o SS3 como prioridade máxima (0).

Registadores

- O registrador **ADCEMUX** controla o gatilho (*trigger*) de cada um dos sequenciadores



- 0x00: O trigger é iniciado setando o bit **SSn** no registrador **ADCPSSI**
- 0x01, 0x02, 0x03: Comparadores Analógicos
- 0x04: Por GPIO
- ...
- 0x0F: Sempre (amostragem contínua)

Registadores

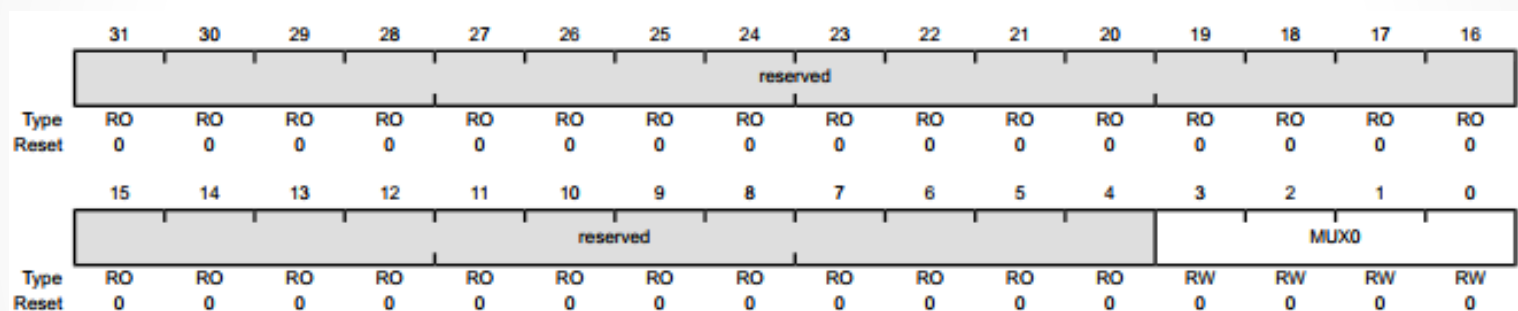
- O registrador **ADCPSSI** permite o *software* inicializar a amostragem nos sequenciadores.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OSYNC	reserved				SYNCAWAIT	reserved									
Type	RW	RO	RO	RO	RW	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved												SS3	SS2	SS1	SS0
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	-

- Os 4 primeiros bits ativam cada um dos sequenciadores.
- Exemplo: para iniciar a amostragem por SW no sequenciador SS3, escrever 0x08.

Registadores

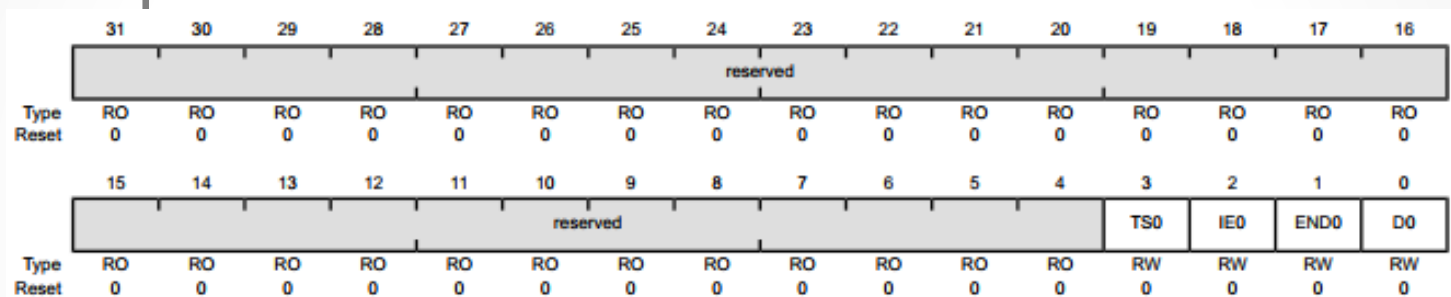
- O registrador **ADCSSMUX3** define qual das 20 entradas analógicas será utilizada para a conversão no sequenciador SS3.



- Os bits de 0 a 3 definem qual AN será utilizado.
- Exemplo: se MUX[3:0] = 2 a entrada analógica a ser utilizada será a AN2
- Obs:** Para utilizar entre a AN16 e AN20 o flag no registrador ADCSSEMUX3 deve ser ativado.

Registadores

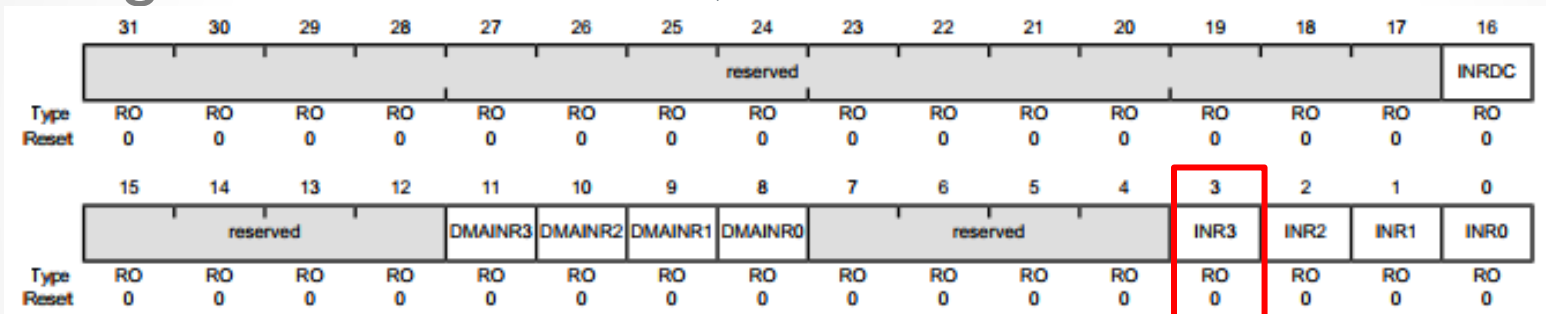
- O registrador **ADCSSCTL3** contém configuração do sequenciador SS3.



- **TS0**: habilita o sensor de temperatura durante a primeira amostra
- **IE0**: Habilita a o sinal de interrupção ser gerado (utilizar se fizer a leitura do registrador **ADCRIS**)
- **END0**: habilitar se uma amostra é o fim da sequência. Como há apenas uma amostra no sequenciador 3, sempre habilitar.
- **D0**: se for entrada diferencial

Registadores

- Quando a conversão estiver completa, o bit do respectivo sequenciador estará ativo no registrador **ADCRIS**, se o bit **IE0** estiver ativado.



- Os 4 primeiros bits mostram se a conversão em cada um dos sequenciadores está pronta.
- Exemplo: se o bit 3 estiver setado a conversão no sequenciador SS3 está pronta

Registradores

- O registrador **ADCISC** contém os bits para limpar o flag de conversão realizada, que pode ser atingido com interrupção ou não. Nele que se faz o ACK da conversão.
- O registrador **ADCIM** contém os bits para ligar a interrupção (se for utilizada)

Passo-a-passo (GPIO)

Para configurar

1. Habilitar o clock no módulo GPIO no registrador **RCGGPIO** (cada bit representa uma GPIO) e esperar até que a respectiva GPIO esteja pronta para ser acessada no registrador **PRGPIO** (cada bit representa uma GPIO).
2. Habilitar a funcionalidade analógica do Pino do GPIO no registrador **GPIOAMSEL**.

Passo-a-passo (GPIO)

3. Escolher o pino do GPIO como entrada no registrador **GPIODIR**
4. Habilitar o bits de função alternativa no registrador **GPIOAFSEL** para o pino do GPIO.
5. Desabilitar a função digital no pino do GPIO no registrador **GPIODEN**.

Passo-a-passo (ADC)

6. Habilitar o clock no módulo ADC no registrador **RCGCADC** (cada bit representa uma ADC) e esperar até que a respectiva ADC esteja pronta para ser acessada no registrador **PRADC** (cada bit representa uma ADC). Escolher ADC0 ou ADC1.
7. Escolher a máxima taxa de amostragem no registrador **ADCPC**.
8. Configurar a prioridade de cada um dos sequenciadores no **ADCSSPRI**. Se utilizar somente um dos sequenciadores (e.g. SS3) não é relevante. Entretanto cada sequenciador deve ter prioridade diferente.

Passo-a-passo (ADC)

9. Desabilitar o sequenciador no registrador **ADCACTSS** para configurá-lo. Por exemplo, para desabilitar o SS3 escrever **0** no bit **ASEN3**.
10. Configurar o tipo de gatilho para cada conversão analógica no registrador **ADCEMUX**. Por exemplo, se for utilizar gatilho por SW no SS3, escrever **0000** nos bits **EM3[3-0]**.
11. Para cada amostra na sequência de amostragem, configurar a fonte de entrada analógica no registrador **ADCSSMUX_n**. Por exemplo, se utilizar o canal **AN0** (PE3) no SS3, escrever **0000** nos bits 3-0 do registrador **ADCSSMUX3**.

Passo-a-passo (ADC)

12. Para cada amostra na sequência de amostragem configurar os bits de controle no nibble correspondente no registrador **ADCSSCTLn**. Sempre o último nibble deve ter o bit **END** setado. Por exemplo, para o SS3, que possui somente uma amostra, habilitar **IE0** e **END0**, escrever **0110** no único nibble do **ADCSSCTL3**.
13. (Opcional) Se utilizar interrupções, setar o bit correspondente no registrador **ADCIM**.
14. Habilitar o sequenciador no registrador **ADCACTSS**. Por exemplo, para habilitar o SS3 escrever **1** no bit **ASEN3**.

Passo-a-passo

Para realizar uma conversão.

1. Fazer uma função separadamente da configuração que seja chamada a cada momento que a conversão seja requerida. Dentro da função fazer os seguintes passos:
2. Iniciar o gatilho de SW, no sequenciador no registrador **ADCPSSI**. Por exemplo, para ativar o SS3 escrever 0x0008.
3. Fazer *polling* do registrador **ADCRIS**, para esperar a conversão. Quando a conversão estiver pronta o bit respectivo é setado.

Passo-a-passo

4. Ler o resultado da conversão no registrador **ADCSSFIFO_n**. Por exemplo, o SS3 só tem uma posição na FIFO então ler somente uma vez o registrador **ADCSSFIFO3** e colocar em uma variável.
5. Realizar o ACK no registrador **ADCISC** para limpar o bit de conversão no registrador **ADCRIS**.
6. Fazer a função retornar o valor lido do ADC que foi colocado na variável.