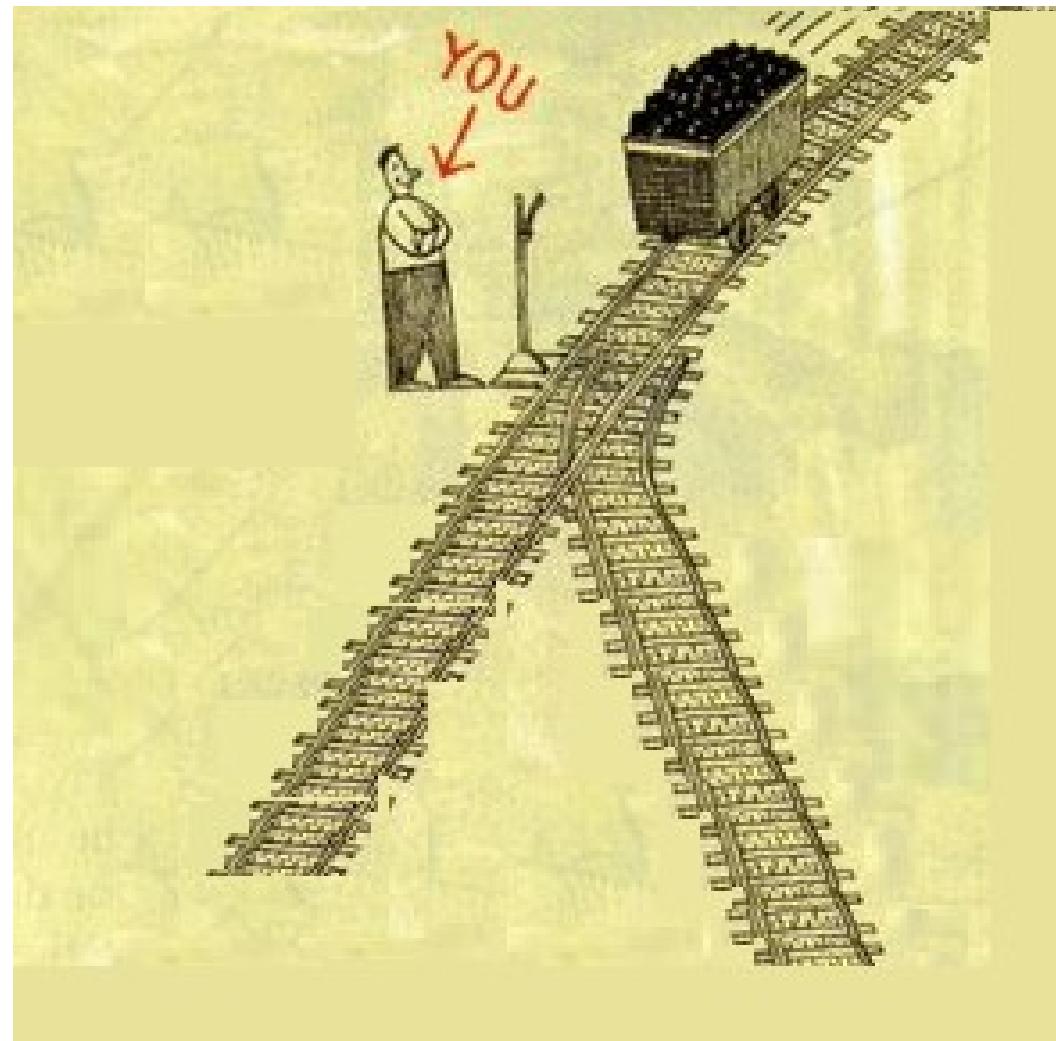


Interrupciones ADC

INTERRUPCIONES

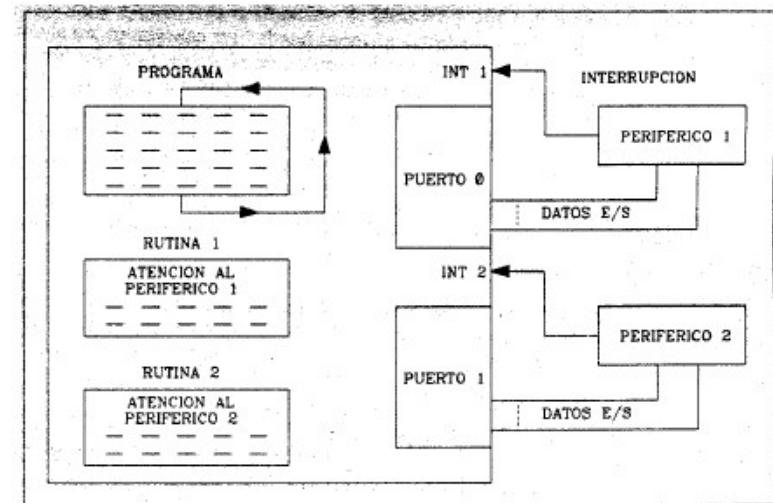
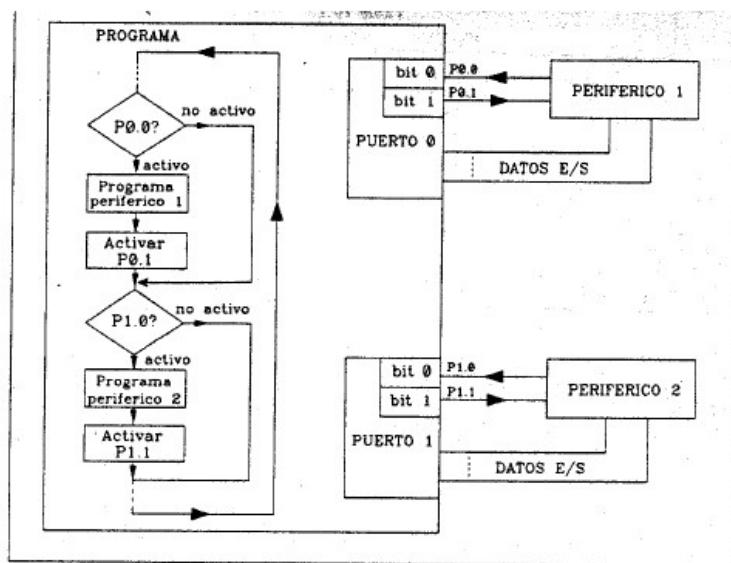


Eventos Asincrónicos Externos

Comunicación asíncrona de sistemas periféricos con la CPU

Encuesta -Consulta
Polling

interrupción



CLASIFICACION DE LAS INTERRUPCIONES

■ NO ENMASCARABLES

No Inhibibles (*non maskarable*)

■ ENMASCARABLES

Inhibibles (*maskarable*)

■ FIJAS

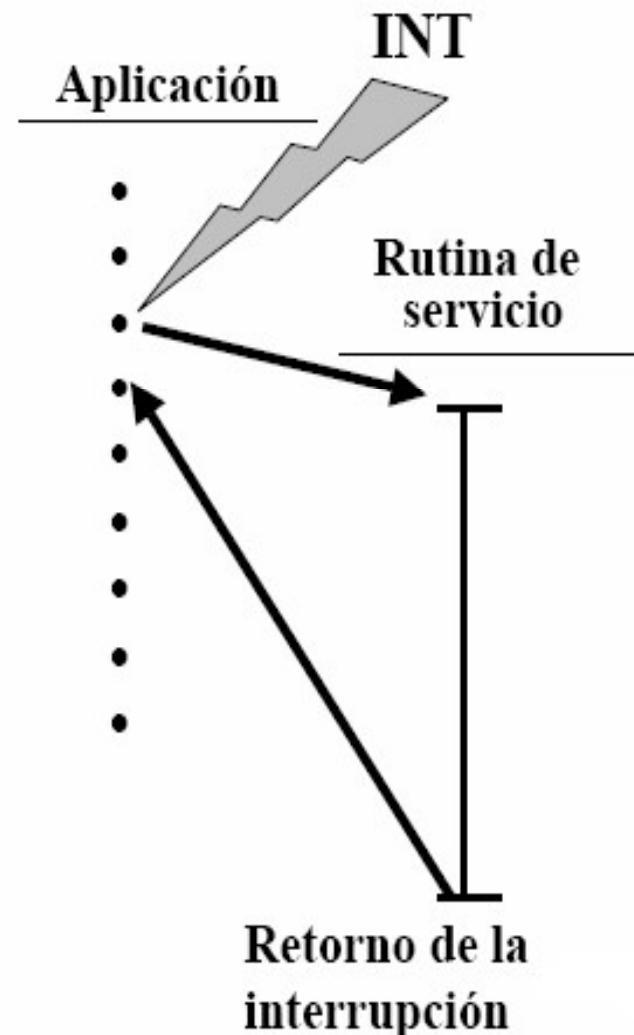
■ VECTORIZADAS

Las fuentes de interrupción son:

- Inicialización o reset.
- Tres interrupciones externas.
- Ocho interrupciones de los timers
(comparación, captura, desbordamiento).
- Al completar una transmisión serial por el SPI.
- Tres por el USART (trans, recep., buffer vacío).
- Termino de conversión ADC.
- Finalización de escritura EEPROM.
- Comparador Analógico
- Interfaz serial TWI.
- Escritura en memoria de programa

Las interrupciones se ejecutan de la siguiente manera

- Ocurre el evento.
- Setea el flag de interrupción.
- Verifica las interrupciones.
- Termina la instrucción que se estaba ejecutando.
- Se guarda la dirección de retorno.
- Carga el PC con la dirección de ISR.
- Desactiva el habilitador global de interrupciones ($I=0$)
- Se desarrolla la interrupción.
- La instrucción de retorno se encarga de activar el habilitador global y obtener la dirección de retorno.



Vectores de interrupción

Vector	Dirección	Fuente	Bandera	Descripción del evento que produce la interrupción
1	\$000	RESET		RESET POR EL TERMINAL, EN EL ENCENDIDO, POR PERDIDA DE Vcc, POR EL WATCHDOG y POR EL JTAG
2	\$002	INT0	INTFO	PEDIDO DE LA INTERRUPCIÓN EXTERNA 0
3	\$004	INT1	INTF1	PEDIDO DE LA INTERRUPCIÓN EXTERNA 1
4	\$006	INT2	INTF2	PEDIDO DE LA INTERRUPCIÓN EXTERNA 2
5	\$008	PCINT0	PCIFO	PEDIDO DE LA INTERRUPCIÓN 0 POR CAMBIO EN LOS TERMINALES
6	\$00A	PCINT1	PCIF1	PEDIDO DE LA INTERRUPCIÓN 1 POR CAMBIO EN LOS TERMINALES
7	\$00C	PCINT2	PCIF2	PEDIDO DE LA INTERRUPCIÓN 2 POR CAMBIO EN LOS TERMINALES
8	\$00E	PCINT3	PCIF3	PEDIDO DE LA INTERRUPCIÓN 3 POR CAMBIO EN LOS TERMINALES
9	\$010	WDT	WDIF	FUERA DE TIEMPO DEL WATCHDOG
10	\$012	TIMER2_COMPA	OCF2A	EMPAREJAMIENTO DEL COMPARADOR A DEL TEMPORIZADOR/CONTADOR 2
11	\$014	TIMER2_COMPB	OCF2B	EMPAREJAMIENTO DEL COMPARADOR B DEL TEMPORIZADOR/CONTADOR 2
12	\$016	TIMER2_OVF	TOV2	DESBORDAMIENTO DEL TEMPORIZADOR/CONTADOR 2
13	\$018	TIMER1_CAPT	ICF1	CAPTURA DEL TEMPORIZADOR/CONTADOR 1
14	\$01A	TIMER1_COMPA	OCF1A	EMPAREJAMIENTO DEL COMPARADOR A DEL TEMPORIZADOR/CONTADOR 1
15	\$01C	TIMER1_COMPB	OCF1B	EMPAREJAMIENTO DEL COMPARADOR B DEL TEMPORIZADOR/CONTADOR 1
16	\$01E	TIMER1_OVF	TOV1	DESBORDAMIENTO DEL TEMPORIZADOR/CONTADOR 1
17	\$020	TIMERO_COMPA	OCFOA	EMPAREJAMIENTO DEL COMPARADOR A DEL TEMPORIZADOR/CONTADOR 0
18	\$022	TIMERO_COMPB	OCFOB	EMPAREJAMIENTO DEL COMPARADOR B DEL TEMPORIZADOR/CONTADOR 0
19	\$024	TIMERO_OVF	TOVO	DESBORDAMIENTO DEL TEMPORIZADOR/CONTADOR 0
20	\$026	SPI_STC	SPIF	TRANSFERENCIA DEL INTERFAZ A PERIFERICOS SERIALES COMPLETA
21	\$028	USART0_RX	RXCO	RECEPCION DEL USART 0 COMPLETA
22	\$02A	USART0_UDRE	UDRE0	REGISTRO DE DATOS DEL USART 0 VACIO
23	\$02C	USART0_TX	TXCO	TRANSMISION DEL USART 0 COMPLETA
24	\$02E	ANALOG_COMP	ACI	COMPARADOR ANALOGICO
25	\$030	ADC	ADIF	CONVERSION COMPLETA DEL CONVERSOR ANALOGICO A DIGITAL (ADC)
26	\$032	EE_READY		EEPROM LISTA PARA UNA NUEVA TRANSFERENCIA
27	\$034	TWI	TWINT	INTERFACE SERIAL CON DOS LINEAS
28	\$036	SPM_READY		ALMACENAMIENTO EN LA MEMORIA DE PROGRAMA COMPLETO
29	\$038	USART1_RX	RXC1	RECEPCION DEL USART 1 COMPLETA
30	\$03A	USART1_UDRE	UDRE1	REGISTRO DE DATOS DEL USART 1 VACIO
31	\$03C	USART1_TX	TXC1	TRANSMISION DEL USART 1 COMPLETA

Vectores de interrupción

- **.ORG 0x000**
- **RJMP Principal** ; Se evita el vector de interrupciones

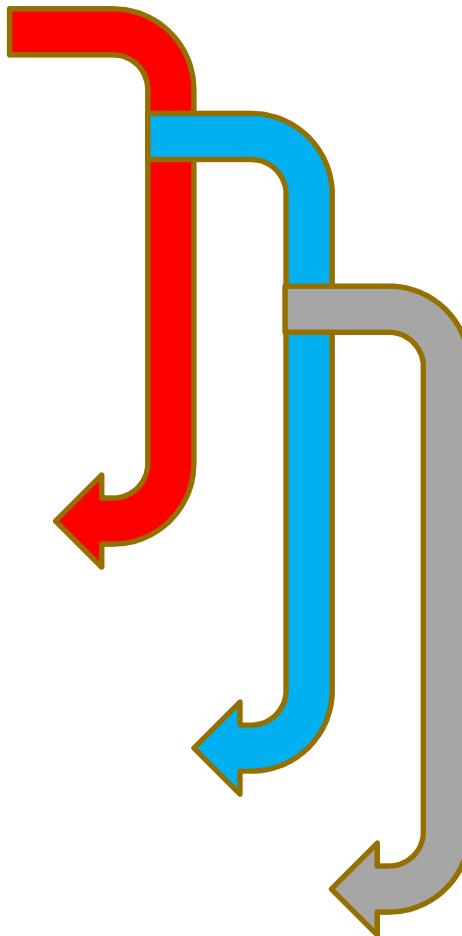
- **.ORG 0x001**
- **RJMP Externa_0** ; Bifurca a su ISR externa 0

- **.ORG 0x002**
- **RJMP Externa_1** ; Bifurca a su ISR externa 1
- ; Si fuera necesario, aquí estarían otras bifurcaciones

- **.ORG 0x013**

- **Principal:** ; Aquí se debe ubicar el código principal
- ; Debe activar las interrupciones
- ; ...
- ; Posterior al código principal, deben situarse las ISRs
- ; Respuesta a la interrupción externa 0
- ; Externa_0:
- **RETI** ; Debe terminar con RETI

- **Externa_1:** ; Respuesta a la interrupción externa 1
- ; ...
- **RETI** ; Debe terminar con RETI



INTERRUPCIONES POR SEÑALES EXTERNAS

Existen tres interrupciones externas y cuatro interrupciones por cambio de estado en los Pórticos de E/S

Vector Dirección	Fuente	Bandera	Descripción del evento que produce la interrupción	
1 \$000	RESET		RESET POR EL TERMINAL, EN EL ENCENDIDO, POR PERDIDA DE Vcc, POR EL WATCHDOG y POR EL JTAG	
2 \$002	INTF0	INT0	PEDIDO DE LA INTERRUPCION EXTERNA 0	
3 \$004	INTF1	INT1	PEDIDO DE LA INTERRUPCION EXTERNA 1	
4 \$006	INTF2	INT2	PEDIDO DE LA INTERRUPCION EXTERNA 2	
5 \$008	PCINT0	PCIFO	PEDIDO DE LA INTERRUPCION 0 POR CAMBIO EN LOS TERMINALES	
6 \$00A	PCINT1	PCIF1	PEDIDO DE LA INTERRUPCION 1 POR CAMBIO EN LOS TERMINALES	
7 \$00C	PCINT2	PCIF2	PEDIDO DE LA INTERRUPCION 2 POR CAMBIO EN LOS TERMINALES	
8 \$00E	PCINT3	PCIF3	PEDIDO DE LA INTERRUPCION 3 POR CAMBIO EN LOS TERMINALES	

INTERRUPCIONES EXTERNAS

**LOS ATMEGA 164P TIENEN TRES
INTERRUPCIONES EXTERNAS**

(PCINT8/XCK0/T0) PB0	1	40	PA0 (ADC0/PCINT0)
(PCINT9/CLK0/T1) PB1	2	39	PA1 (ADC1/PCINT1)
(PCINT10/INT2/AIN0) PB2	3	38	PA2 (ADC2/PCINT2)
(PCINT11/OC0A/AIN1) PB3	4	37	PA3 (ADC3/PCINT3)
(PCINT12/OC0B/S5) PB4	5	36	PA4 (ADC4/PCINT4)
(PCINT13/MOSI) PB5	6	35	PA5 (ADC5/PCINT5)
(PCINT14/MISO) PB6	7	34	PA6 (ADC6/PCINT6)
(PCINT15/SCK) PB7	8	33	PA7 (ADC7/PCINT7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL1	12	29	PC7 (TOSC2/PCINT23)
XTAL2	13	28	PC6 (TOSC1/PCINT22)
(PCINT24/RXD0) PD0	14	27	PC5 (TDI/PCINT21)
(PCINT25/TXD0) PD1	15	26	PC4 (TDO/PCINT20)
(PCINT26/RXD1/INT0) PD2	16	25	PC3 (TMS/P/CINT19)
(PCINT27/TXD1/INT1) PD3	17	24	PC2 (TCK/P/CINT18)
(PCINT28/XCK1/OC1B) PD4	18	23	PC1 (SDA/P/CINT17)
(PCINT29/OC1A) PD5	19	22	PC0 (SCL/P/CINT16)
(PCINT30/OC2B/ICP) PD6	20	21	PD7 (OC2A/PCINT31)

INT0 TERMINAL PD.2
INT1 TERMINAL PD.3
INT2 TERMINAL PB.2

**POR ESTOS TERMINALES, QUE SE LOS
CONFIGURA COMO ENTRADAS, SE
INGRESAN LAS SEÑALES QUE
PRODUCEN INTERRUPCIONES**

Registros del ADC

- EIMSK = External interrupts Mask Register.
Contiene los bits enable (habilitación)
- EIFR = External Interrupts Flags Register.
Contiene los bits de flag.
- EICRA = External Interrupts Control Register
A. Configura la señal externa que va a generar la interrupción. Es A porque hay AVR más grandes con más interrupciones INTx donde además existe el registro EICRB.

MODOS DE OPERACIÓN DE LAS INTERRUPCIONES EXTERNAS

ISCn1	ISCn0	DESCRIPCIÓN
0	0	El nivel lógico 0 en la entrada INTn genera un pedido de interrupción
0	1	Cualquier cambio en la entrada INTn genera un pedido de interrupción
1	0	La transición de bajada en la entrada INTn genera un pedido de interrupción
1	1	La transición de subida en la entrada INTn genera un pedido de interrupción

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
EICRA	-	-	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00		(\$69)

Control de interrupciones

■ Bit enable

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
EIMSK	-	-	-	-	-	INT2	INT1	INT0	\$1D	(\$3D)

■ Bit flag

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
EIFR	-	-	-	-	-	INTF2	INTF1	INTFO	\$1C	(\$3C)

■ Bit enable general

SREG – AVR Status Register

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0		SREG
0x3F (0x5F)	I	T	H	S	V	N	Z	C		
Read/Write	R/W									
Initial Value	0	0	0	0	0	0	0	0		

HABILITACIÓN DE LAS INTERRUPCIONES EXTERNAS

LOS BITS INT0, INT1 e INT2 EN 1 HABILITAN LOS PEDIDOS INDIVIDUALES DE INTERRUPCIÓN, CAUSADOS POR LA ACTIVIDAD EN LA RESPECTIVA ENTRADA DE LA SEÑAL

PARA QUE SURTAN EFECTO ESTOS PEDIDOS TAMBIÉN DEBE ESTAR EN 1 LA BANDERA GENERAL DE HABILITACIÓN DE INTERRUPCIONES

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
EIMSK	-	-	-	-	-	INT2	INT1	INT0	\$1D	(\$3D)

BANDERAS DE LAS INTERRUPCIONES EXTERNAS

**INTF0, INTF1 e INTF2 SE PONEN EN 1
POR EFECTO DE LA ACTIVIDAD DE LA
ENTRADA DE LA SEÑAL RESPECTIVA
REGRESAN A 0 CUANDO SE EJECUTA
LA RUTINA DE Interrupción O
CUANDO SE ESCRIBE UN 1 LÓGICO**

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
EIFR	-	-	-	-	-	INTF2	INTF1	INTF0	\$1C	(\$3C)

0x35	7	6	5	4	3	2	1	0	MCUCR
	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISCO0	

ISCx1	ISCx0	Activación de la Interrupción
0	0	Por un nivel bajo de voltaje en INTx
0	1	Por cualquier cambio lógico en INTx
1	0	Por un flanco de bajada en INTx
1	1	Por un flanco de subida en INTx

0x34	7	6	5	4	3	2	1	0	MCUCSR
	JTD	ISC2	-	JTRF	WDRF	BORF	EXTRF	PORF	

ISC2	Activación de la Interrupción
0	Por un flanco de bajada en INT2
1	Por un flanco de subida en INT2

0x3B	7	6	5	4	3	2	1	0	GICR
	INT1	INT0	INT2	-	-	-	IVSEL	IVCE	

Bit 7 – INT1: Habilitador individual de la interrupción externa 1

Bit 6 – INT0: Habilitador individual de la interrupción externa 0

Bit 5 – INT2: Habilitador individual de la interrupción externa 2

0x3B	7	6	5	4	3	2	1	0	GICR
	INT1	INT0	INT2	-	-	-	IVSEL	IVCE	

- Bit 7 – INT1: Habilitador individual de la interrupción externa 1**

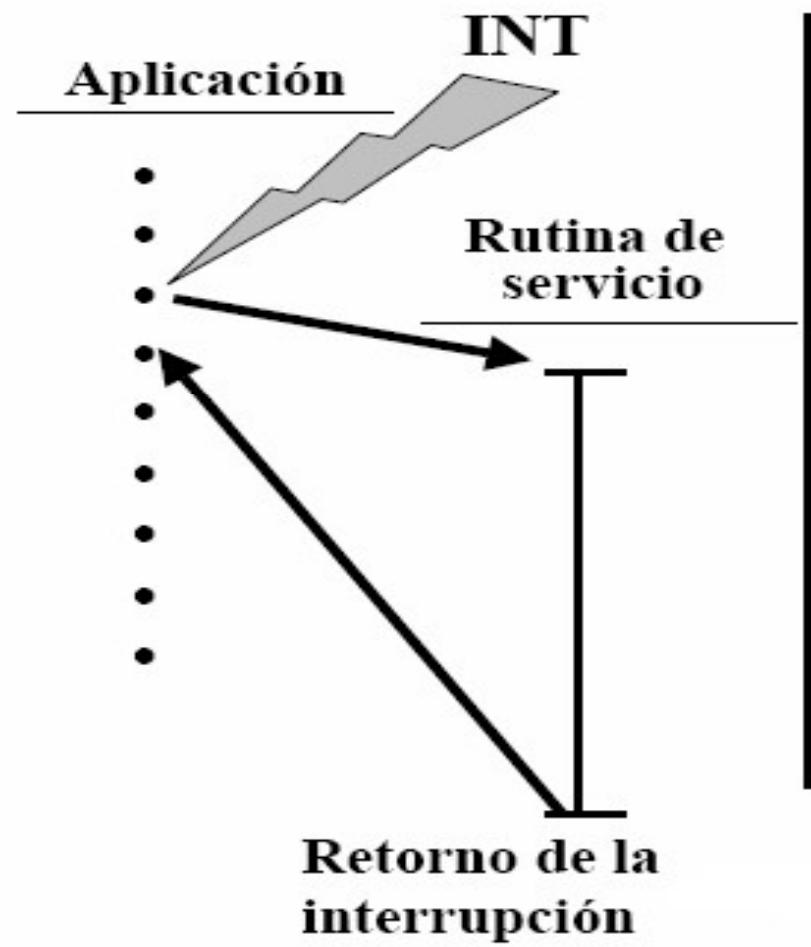
- Bit 6 – INT0: Habilitador individual de la interrupción externa 0**

- Bit 5 – INT2: Habilitador individual de la interrupción externa 2**

**ATMEGA 8
ATMEGA 16**

Control de interrupciones

- El hardware limpia automáticamente el bit de Flag apenas se empiece a ejecutar la función de interrupción
- Los flags se habilitan independientemente de si las interrupciones están habilitadas
- Al ejecutarse la función de interrupción también se limpia por hardware el **bit enable general** I para evitar que se disparen otras interrupciones cuando se esté ejecutando la interrupción actual.
- Sin embargo, la arquitectura de los AVR le permite soportar ese tipo de interrupciones, llamadas recurrentes o anidadas, y si así lo deseamos podemos setear en el bit I dentro de la ISR actual
- SEI CLI



SEI

CLI

SEI Habilitar las interrupciones globales

Sintaxis: SEI

Operación: I \leftarrow 1

Operandos: Ninguno

Descripción: Pone a uno la bandera de habilitación de interrupciones globales I del registro de estado SREG.

Banderas afectadas:

I: Se pone a uno

Ejemplo:

```
SEI          ; Interrupciones habilitadas  
SLEEP        ; Activación de modo SLEEP, esperando  
              ; alguna interrupción
```

CLI Deshabilitar las interrupciones globales

Sintaxis: CLI

Operación: N \leftarrow 0

Operandos: Ninguno

Descripción: Pone a cero la bandera de habilitación de interrupciones globales I del registro de estado SREG.

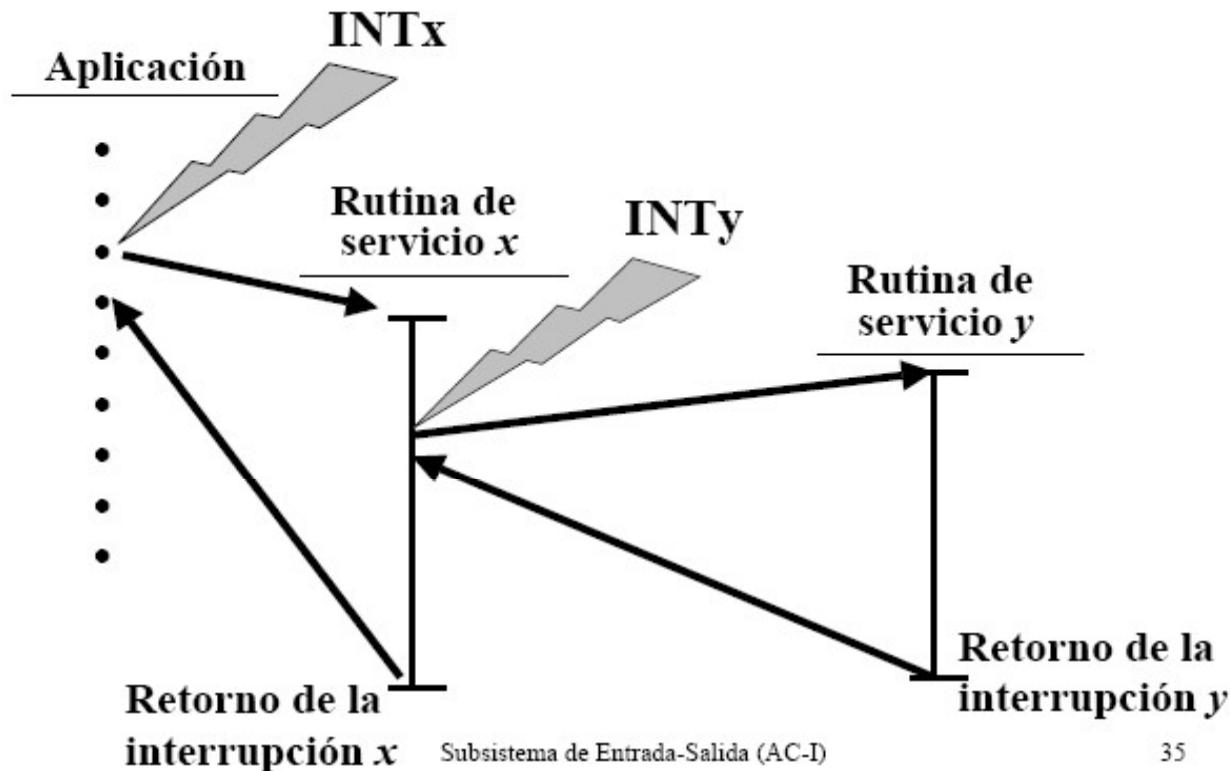
Banderas afectadas:

I: Se pone a cero

Ejemplo:

```
CLI          ; Interrupciones deshabilitadas  
IN r11, $16   ; Leer el Puerto B  
SEI          ; Interrupciones habilitadas
```

Interrupciones anidadas



35

Al ejecutarse la función de interrupción también se limpia por hardware el **bit enable general I** para evitar que se disparen otras interrupciones cuando se esté ejecutando la interrupción actual

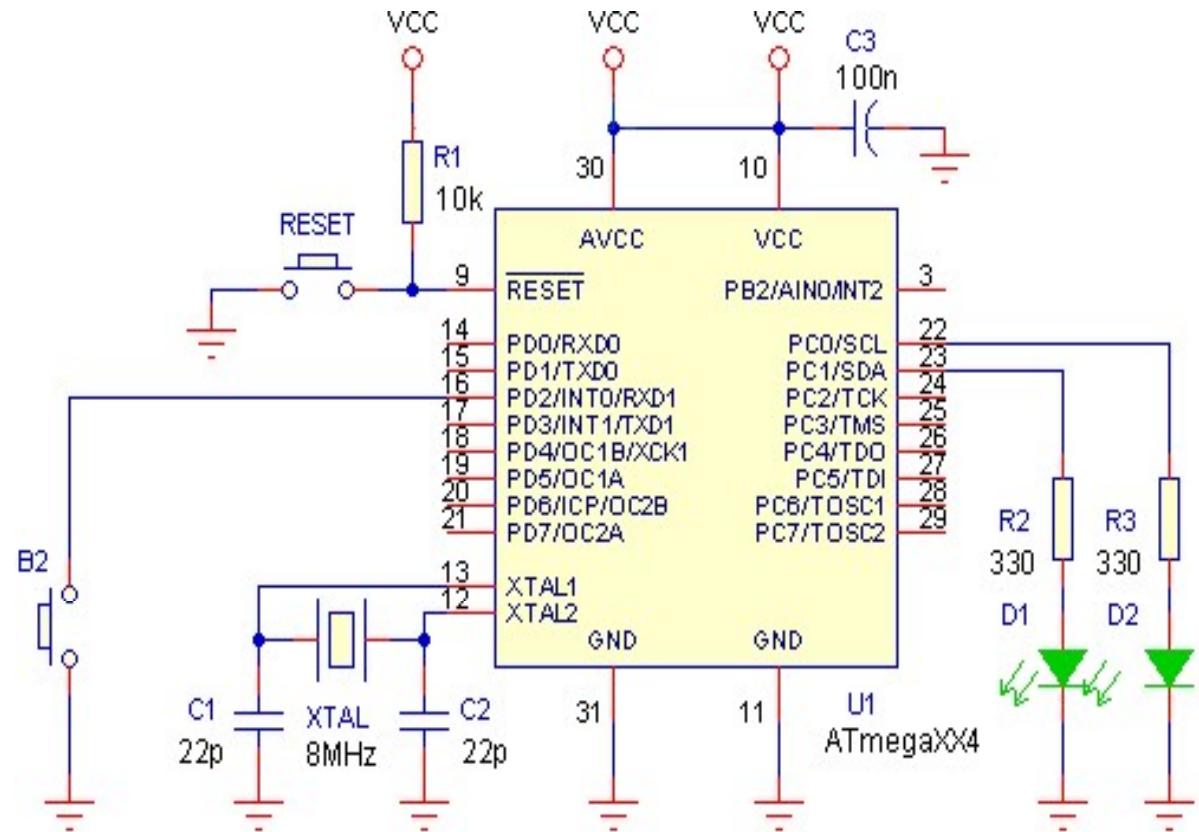
¿NO Permite el uso
de Interrupciones
anidadas ?

→

SI Hay que habilitar Ig

Ejemplo

ISCn1	ISCn0	DESCRIPCIÓN
0	0	El nivel lógico 0 en la entrada INTn genera un pedido de interrupción
0	1	Cualquier cambio en la entrada INTn genera un pedido de interrupción
1	0	La transición de bajada en la entrada INTn genera un pedido de interrupción
1	1	La transición de subida en la entrada INTn genera un pedido de interrupción



Que falta?

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
EIFR	-	-	-	-	-	INTF2	INTF1	INTFO	\$1C	(\$3C)

SEI

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
EICRA	-	-	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00		(\$69)

ISCn1	ISCn0	DESCRIPCIÓN
0	0	El nivel lógico 0 en la entrada INTn genera un pedido de interrupción
0	1	Cualquier cambio en la entrada INTn genera un pedido de interrupción
1	0	La transición de bajada en la entrada INTn genera un pedido de interrupción
1	1	La transición de subida en la entrada INTn genera un pedido de interrupción

; PROGRAMACIÓN DE LAS INTERRUPCIONES EXTERNAS

```

LDI      AUX1, 0B 00101111
STS      EICRA, AUX1          ; INT2 T. POS. INT1/0 T. NEGAT.
LDI      AUX1, 0B00000100
OUT     EIMSK, AUX1          ; INT2 HABILITADA
LDI      AUX1, 0B00000111
OUT     EIFR, AUX1           ; BORRAR LAS BANDERAS

```

; VALORES INICIALES DE HH:MM:SS

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
EIMSK	-	-	-	-	-	INT2	INT1	INT0	\$1D	(\$3D)
REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
EIFR	-	-	-	-	-	INTF2	INTF1	INTFO	\$1C	(\$3C)
REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
EICRA8	-	-	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00		(\$69)

Ejemplo

INTERRUPCIONES POR CAMBIOS EN LOS PÓRTICOS

**LOS ATMEGA 164P TIENEN CUATRO
INTERRUPCIONES PARA LOS CAMBIOS
EN LOS PUERTOS**

PCINT0 PARA EL PUERTO A

PCINT1 PARA EL PUERTO B

PCINT2 PARA EL PUERTO C

PCINT3 PARA EL PUERTO D

**QUE TAMBIÉN SE LOS CONFIGURA COMO
ENTRADAS**

HABILITACIÓN DE LAS INTERRUPCIONES POR CAMBIO

LOS BITS PCIE0, PCIE1, PCIE2 y PCIE3 EN 1 HABILITAN LOS PEDIDOS DE INTERRUPCIÓN, CAUSADOS POR LOS CAMBIOS EN LAS ENTRADAS DE LOS PÓRTICOS A, B, C y D RESPECTIVAMENTE

PARA QUE SURTAN EFECTO ESTOS PEDIDOS TAMBIÉN DEBE ESTAR EN 1 LA BANDERA GENERAL DE HABILITACIÓN DE INTERRUPCIONES

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
PCICR	-	-	-	-	PCIE3	PCIE2	PCIE1	PCIE0		(\$68)

BANDERAS DE LAS INTERRUPCIONES POR CAMBIO

**PCIF0, PCIF1, PCIF2 y PCIF3 SE PONEN EN 1
POR EFECTO DE CAMBIOS PRODUCIDOS EN
LAS ENTRADAS DE LOS PÓRTICOS A, B, C y D
RESPECTIVAMENTE
REGRESAN A 0 CUANDO SE EJECUTA LA
RUTINA DE INTERRUPCIÓN O CUANDO SE
ESCRIBE UN 1 LÓGICO**

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
PCIFR	-	-	-	-	PCIF3	PCIF2	PCIF1	PCIFO	\$1B	(\$3B)

IDENTIFICACIÓN DE LAS ENTRADAS

LA DENOMINACIÓN QUE TIENE CADA UNO DE LOS TERMINALES DE LOS PUERTOS COMO INTERRUPCIONES ES:

PCINT7..0 PARA LOS DEL PUERTO A

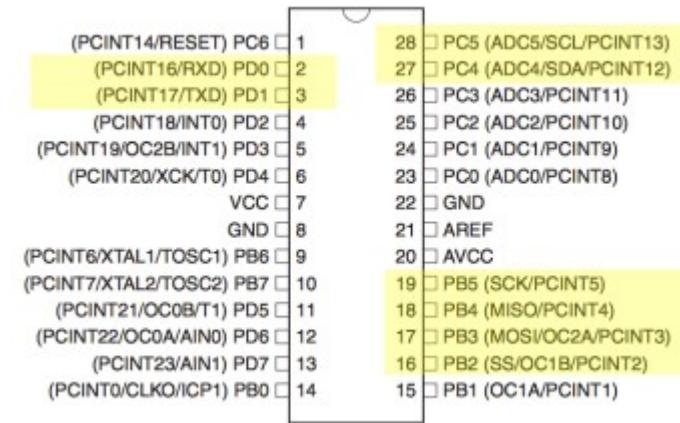
PCINT15..8 PARA LOS DEL PUERTO B

PCINT23..16 PARA LOS DEL PUERTO C

PCINT31..24 PARA LOS DEL PUERTO D

MASCARA DE LAS INTERRUPCIONES POR CAMBIO

**LOS BITS PCINT7..0, PCINT15..8, PCINT23..16 y
PCINT31..24 EN 1 HABILITAN
INDIVIDUALMENTE LAS ENTRADAS DE LOS
PÓRTICOS A, B, C y D RESPECTIVAMENTE
PARA QUE LOS CAMBIOS GENEREN PEDIDOS
DE INTERRUPCIÓN**



REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
PCMSK3	PCINT31	PCINT30	PCINT29	PCINT28	PCINT27	PCINT26	PCINT25	PCINT24		(\$73)
PCMSK2	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16		(\$6D)
PCMSK1	PCINT15	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8		(\$6C)
PCMSK0	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0		(\$6B)

A D C



$$\text{CODIGO DE SALIDA DIGITAL} = \frac{\text{Entrada Analógica}}{\text{Entrada de referencia}} \times (2^N - 1)$$

- Genera un valor binario de salida proporcional al cociente entre la señal de entrada y la de referencia
- Posibles valores binarios de salida: 2^N
- Convierte una señal continua en el tiempo en valores discretos tomados con una determinada frecuencia de muestreo



Los amplificadores operacionales pueden realizar todas las tareas propias del acondicionador de señal:

- **Amplificación/atenuación**: escalado de la señal de entrada para cumplir el margen dinámico del *ADC*
- **Tensión de offset**: desplazar el nivel DC de la señal de entrada para ajustarlo al rango de operación del *ADC*
- **Filtrado**: eliminación de las componentes frecuenciales indeseables de la señal de entrada para ajustar su ancho de banda al del *ADC*

Que es ADC? ¿Cual es su uso?

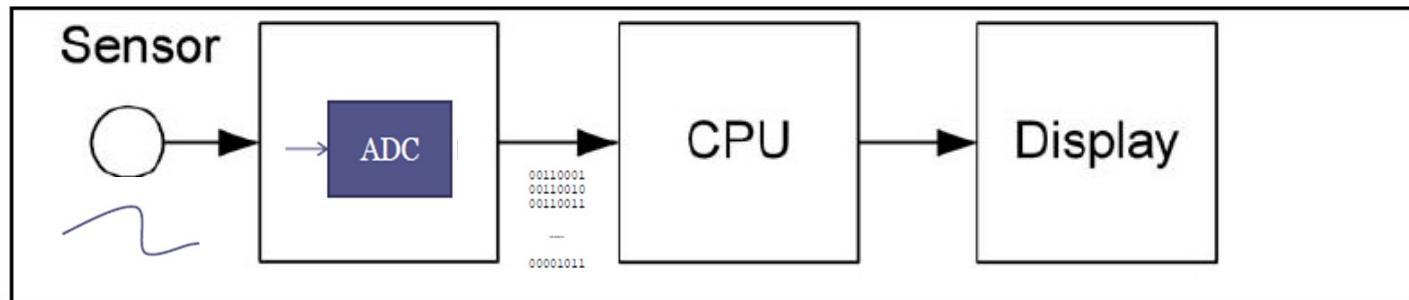
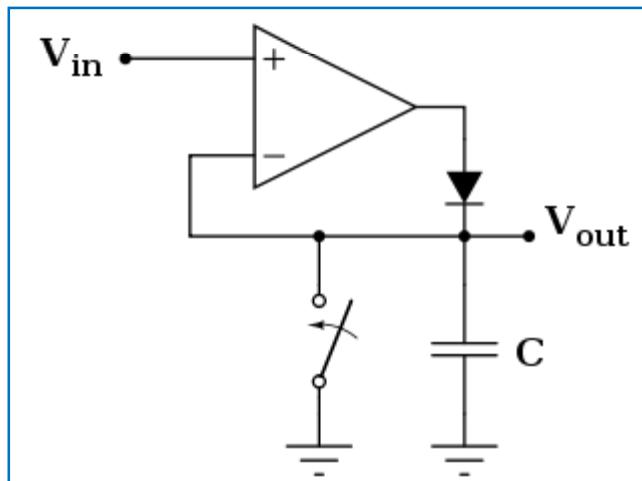


Figure 13-1. Microcontroller Connection to Sensor via ADC



Muestreo: el muestreo (en inglés, sampling) consiste en tomar muestras periódicas de la amplitud de onda. La velocidad con que se toma esta muestra, es decir, el número de muestras por segundo, es lo que se conoce como frecuencia de muestreo.

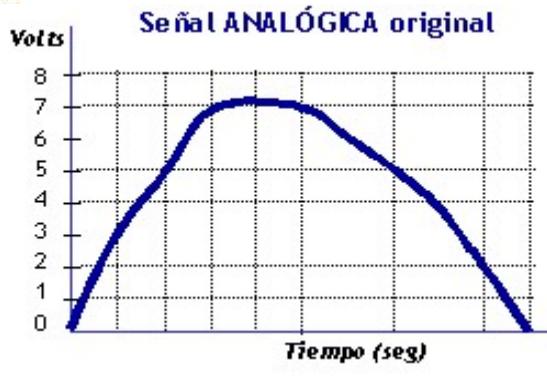
Retención (en inglés, hold): las muestras tomadas han de ser retenidas (retención) por un circuito de retención (hold), el tiempo suficiente para permitir evaluar su nivel (cuantificación). Desde el punto de vista matemático este proceso no se contempla, ya que se trata de un recurso técnico debido a limitaciones prácticas, y carece, por tanto, de modelo matemático.

Cuantificación: en el proceso de cuantificación se mide el nivel de voltaje de cada una de las muestras. Consiste en asignar un margen de valor de una señal analizada a un único nivel de salida. Incluso en su versión ideal, añade, como resultado, una señal indeseada a la señal de entrada: el ruido de cuantificación.

Codificación: la codificación consiste en traducir los valores obtenidos durante la cuantificación al código binario. Hay que tener presente que el código binario es el más utilizado, pero también existen otros tipos de códigos que también son utilizados.

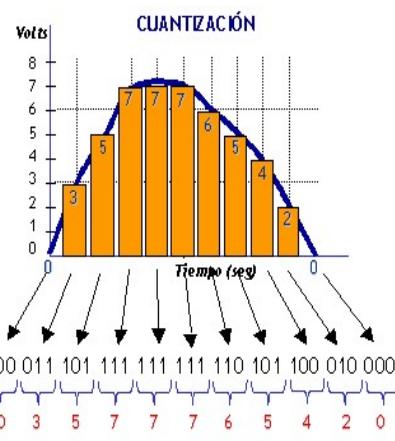
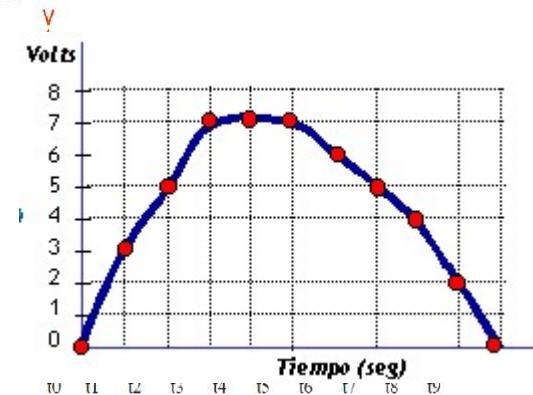
Señal analógica

- Una señal analógica toma un continuo de valores a lo largo del tiempo:

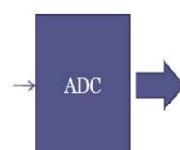
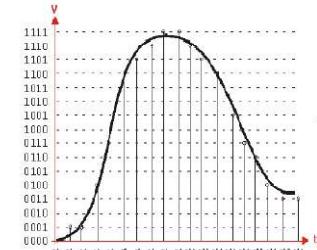


Muestreo

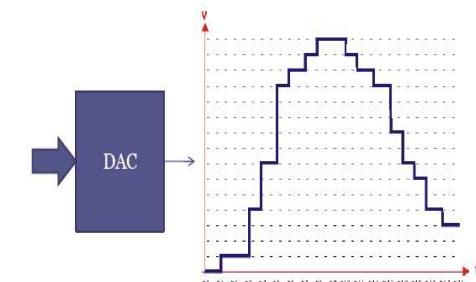
- Para digitalizar una señal analógica se deben tomar muestras a intervalos periódicos de tiempo, conocido como Periodo de Muestreo:



Convertidor analógico a digital

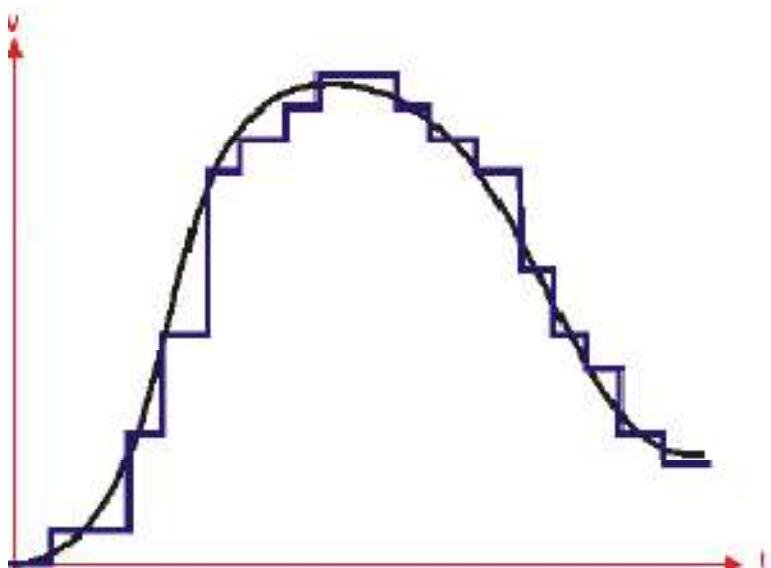


Convertidor digital a analógico

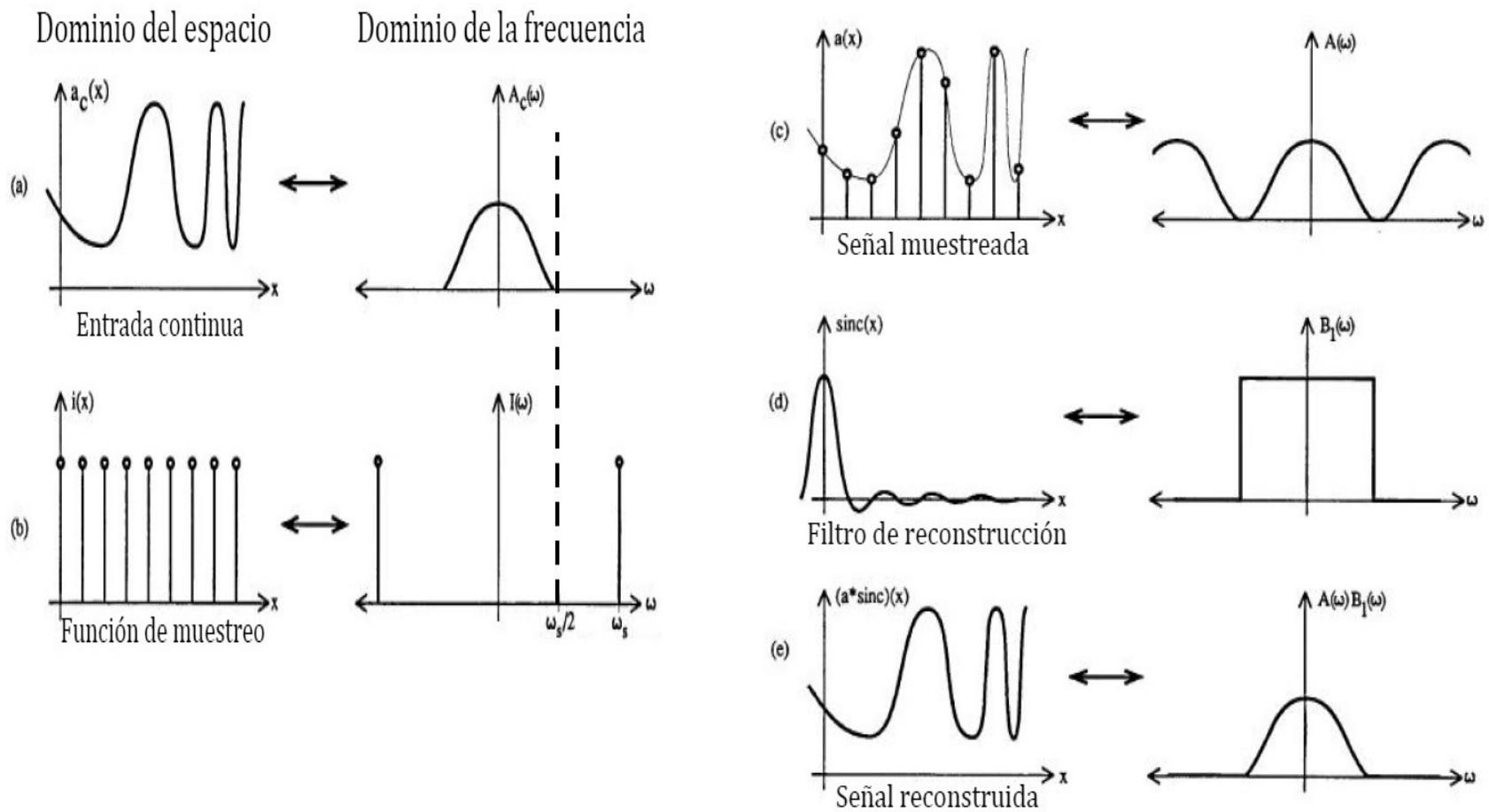


Conversión y recuperación de la información

- Para una adecuada conversión y recuperación de la información se requiere que:
- La resolución del ADC proporcione el número de combinaciones suficientes, de acuerdo con lo que se desea representar.
- La frecuencia de muestreo debe ser por lo menos el doble de la frecuencia de la señal analógica.

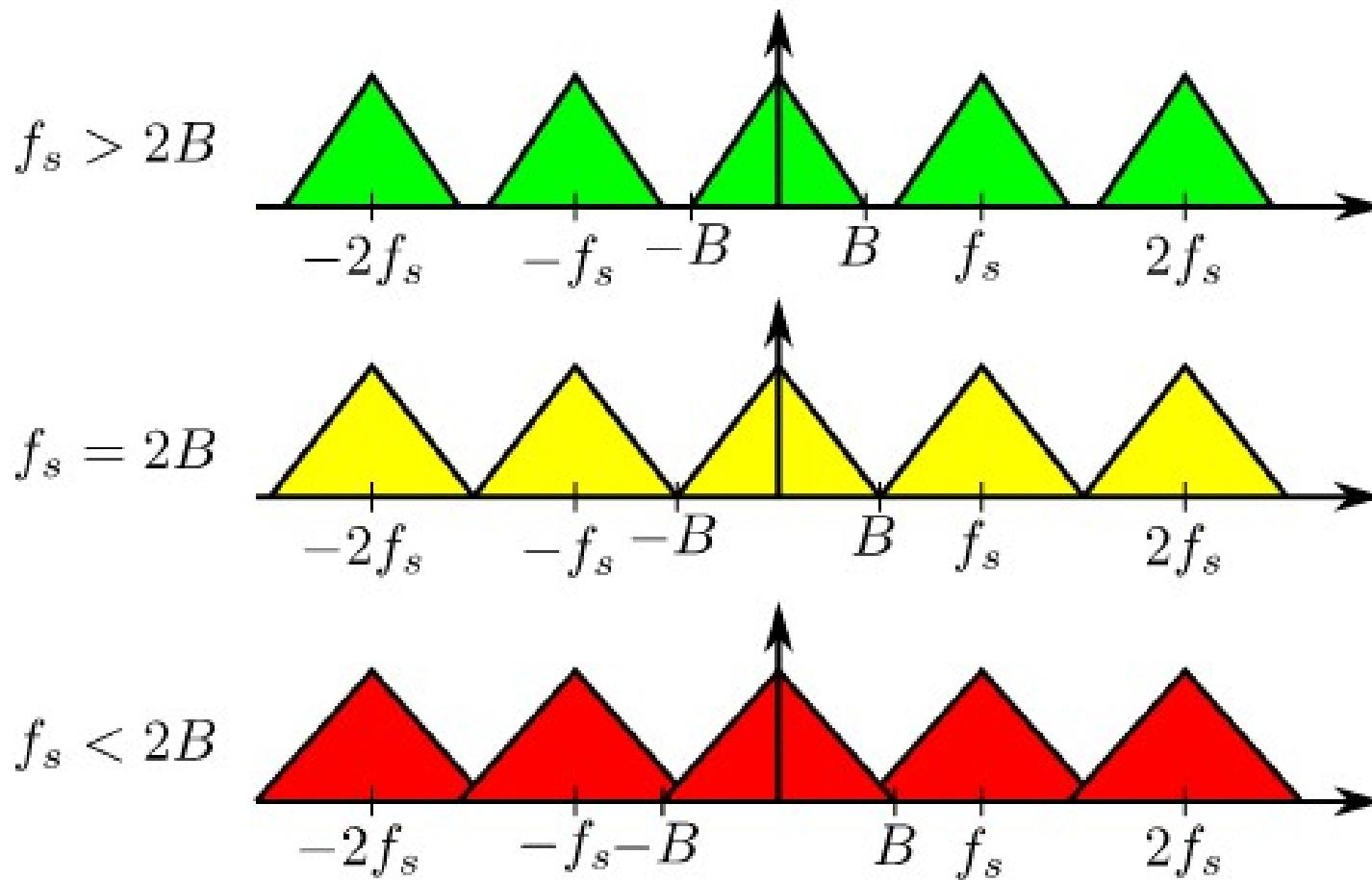


Teorema de muestreo Nyquist



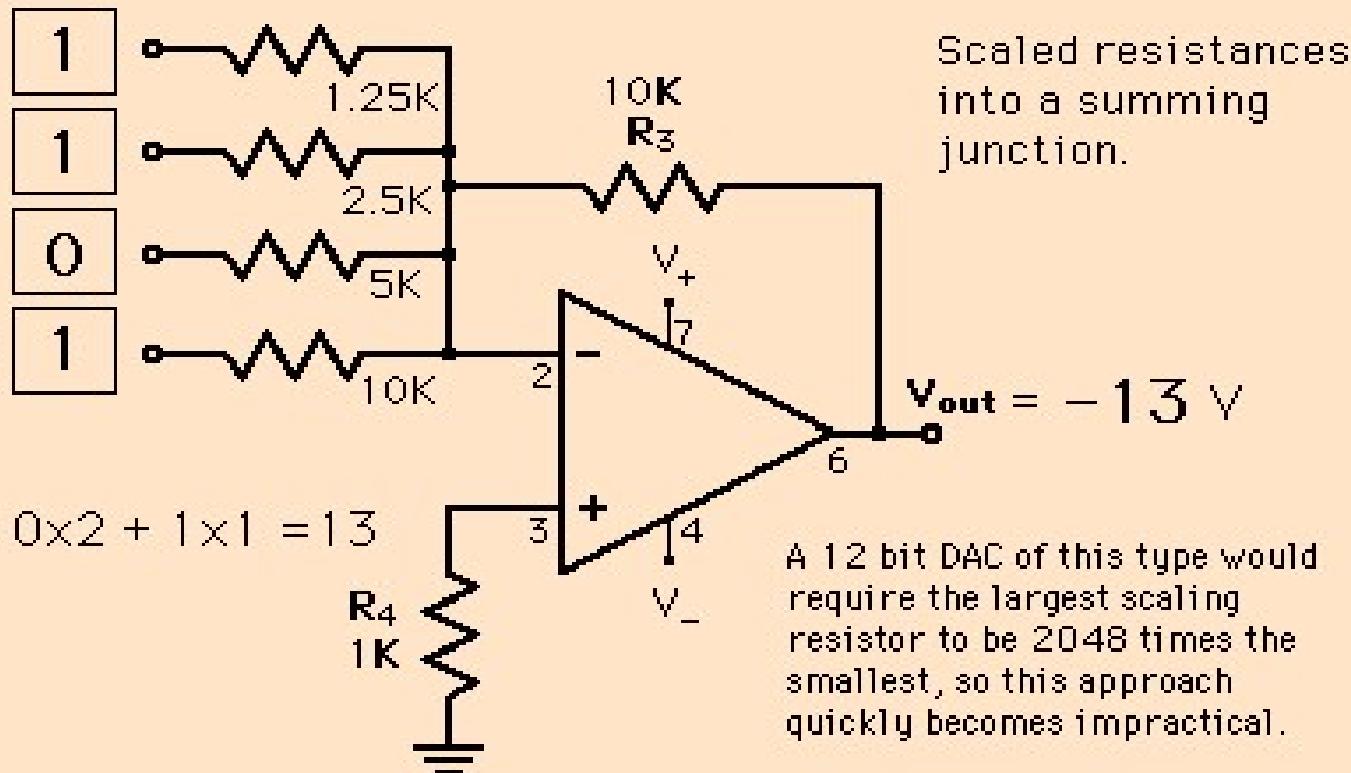
Teorema de muestreo Nyquist

Aliasing



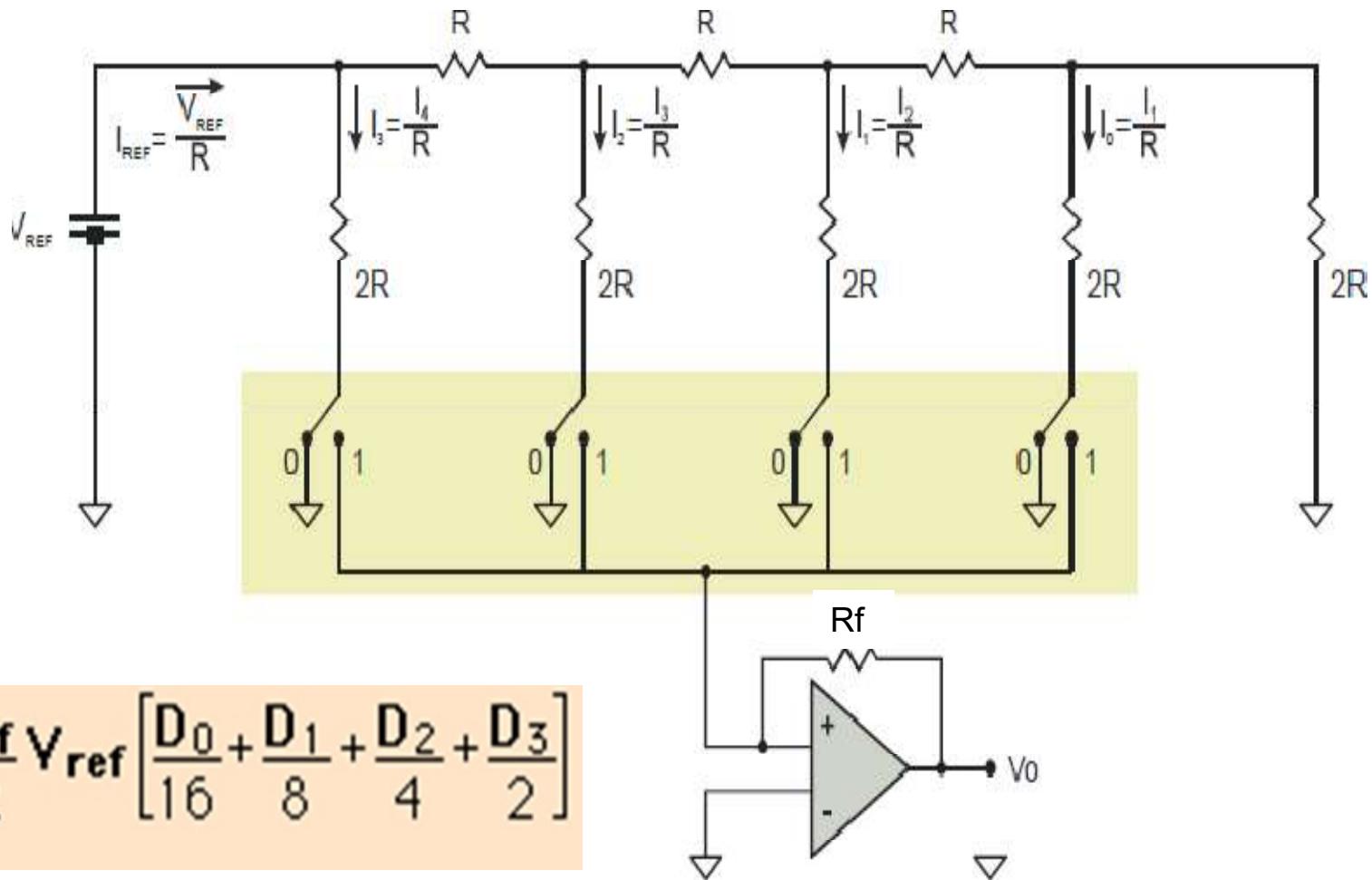
Convertidor D/A de 4 Bits

Inputs in volts are weighted in the summing amplifier to produce the corresponding analog voltage.



Este método no es satisfactorio para un gran número de bits, porque se requiere demasiada precisión en las resistencias sumadoras. Este problema se resuelve con el DAC red R-2R

DAC con base en una Red R - 2R

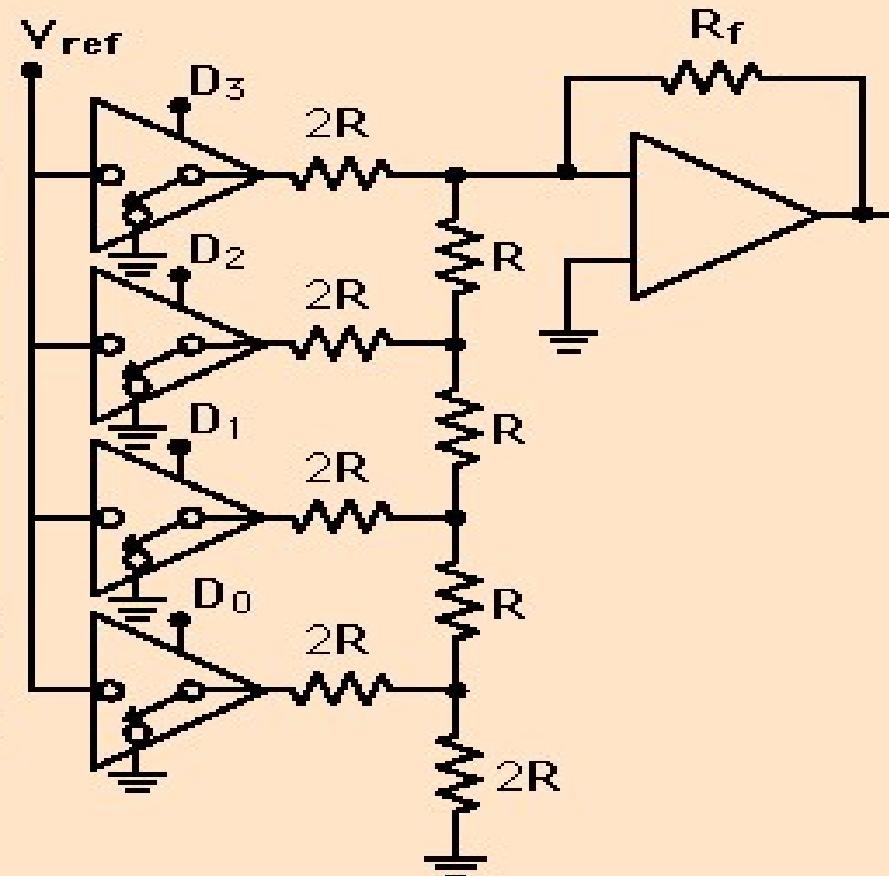


DAC Escalera R-2R

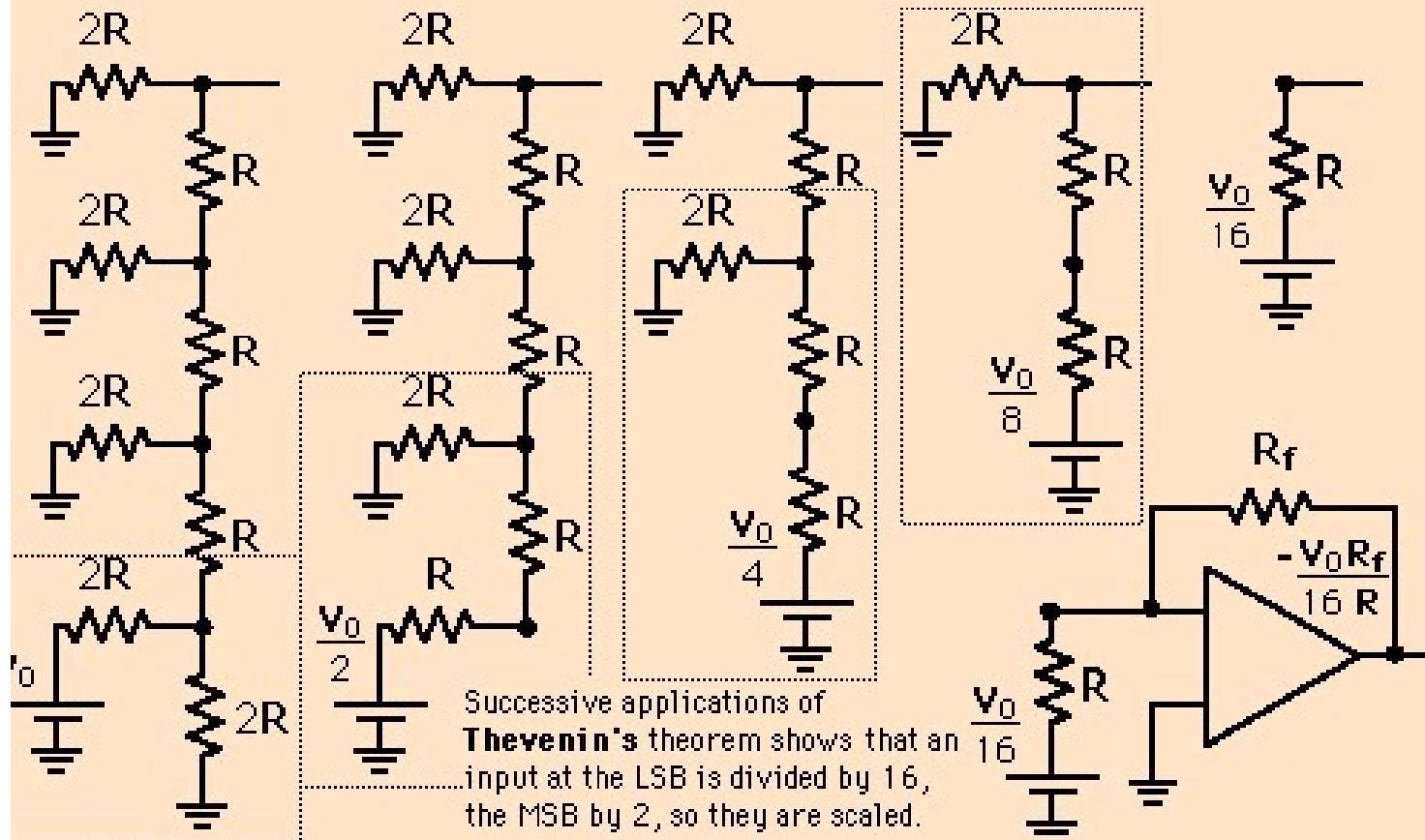
El amplificador sumador con la escalera de resistencias R-2R mostrado produce la salida

$$V_{out} = \frac{R_f}{R} V_{ref} \left[\frac{D_0}{16} + \frac{D_1}{8} + \frac{D_2}{4} + \frac{D_3}{2} \right]$$

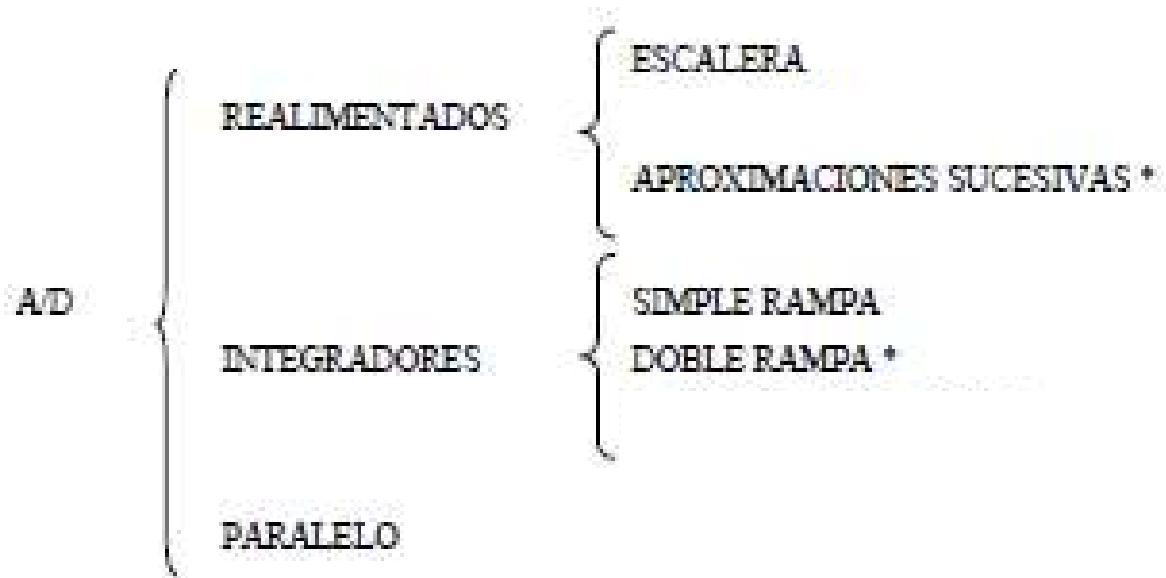
donde las Ds toman los valores 0 ó 1. Estas entradas digitales podrían ser voltajes TTL, los cuales cierran los interruptores con el 1 lógico y los deja en su estado básico de tierra para la lógica 0. Esto se ilustra para 4 bits, pero se puede extender a cualquier número con solo los valores de resistencias R y 2R.



Ejemplo Thevenin: Escalera R-2R



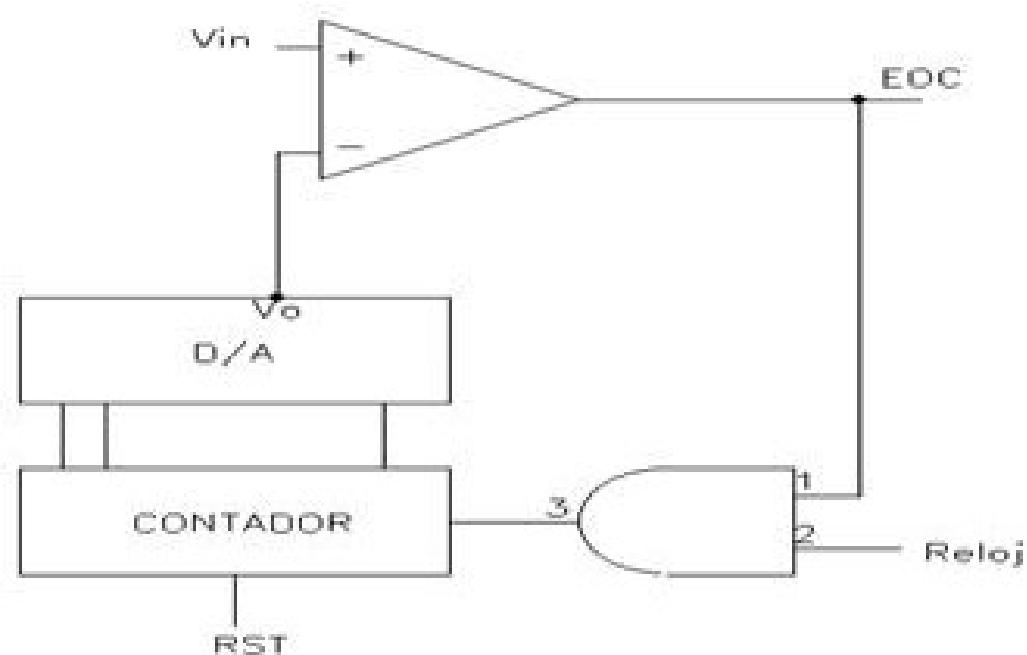
Clasificación de los ADC



ADC Escalera

Si $V_+ > V_-$ sat. Positiva y $V_o = 5V$

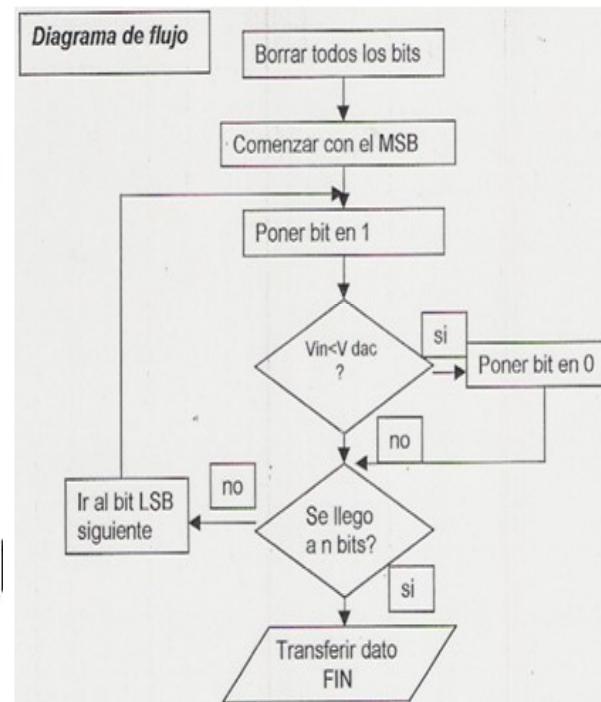
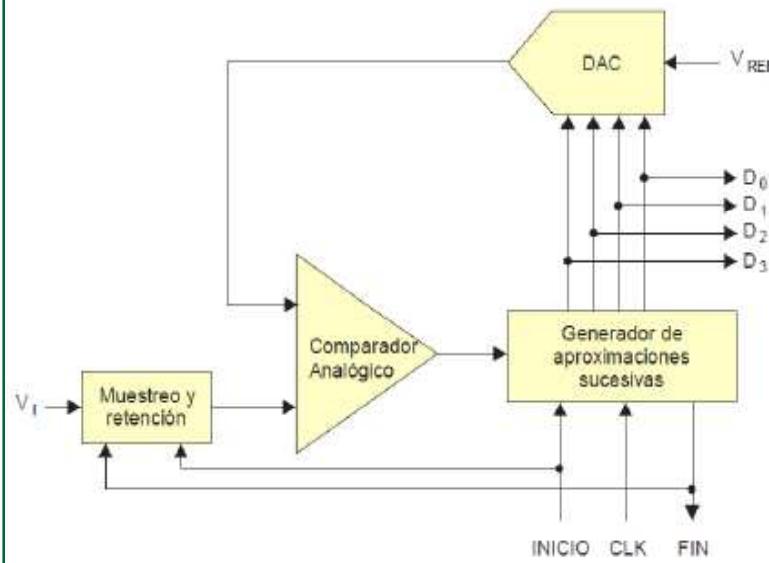
Si $V_+ < V_-$ sat. Negativa y $V_o = 0V$



$$t_{out} = T_s \frac{1}{f_{out}}$$

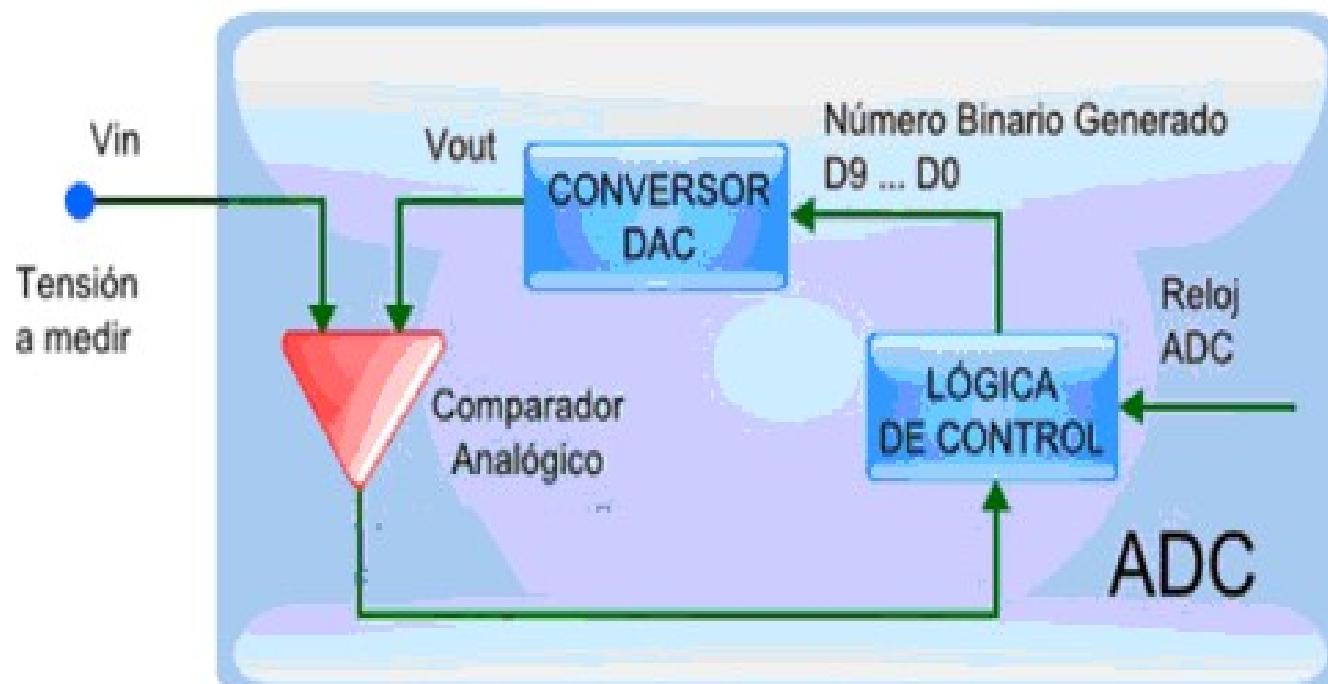
- Arquitectura idónea para resoluciones medias-altas y frecuencia de muestreo menor de 5 MS/s
- Resolución de 8 a 16 bits
- Bajo consumo

ADC de Aproximaciones Sucesivas



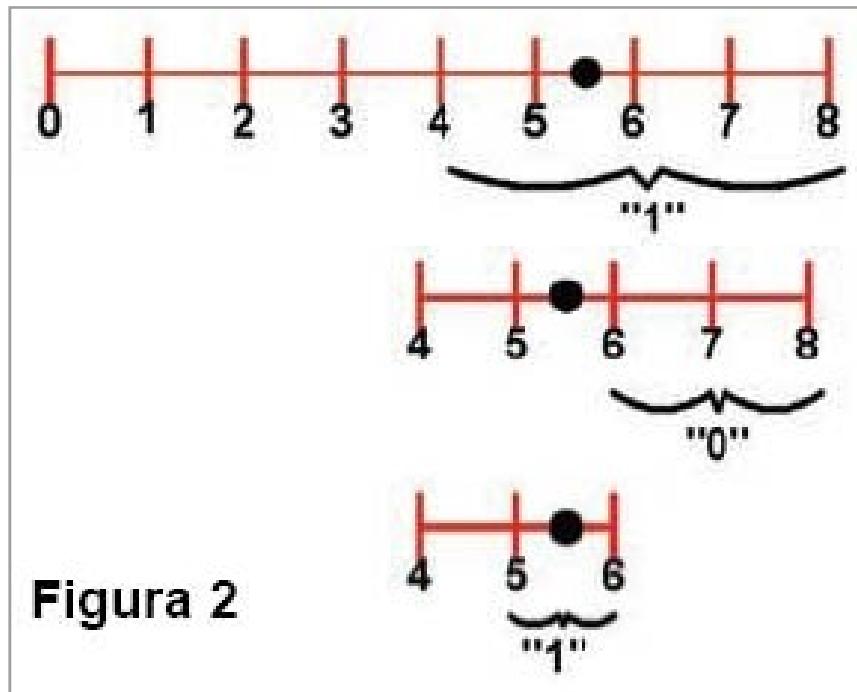
Un modo de carrera libre se consigue cuando el FIN se conecta con el INICIO.

ADC DE APROXIMACIONES SUCESIVAS



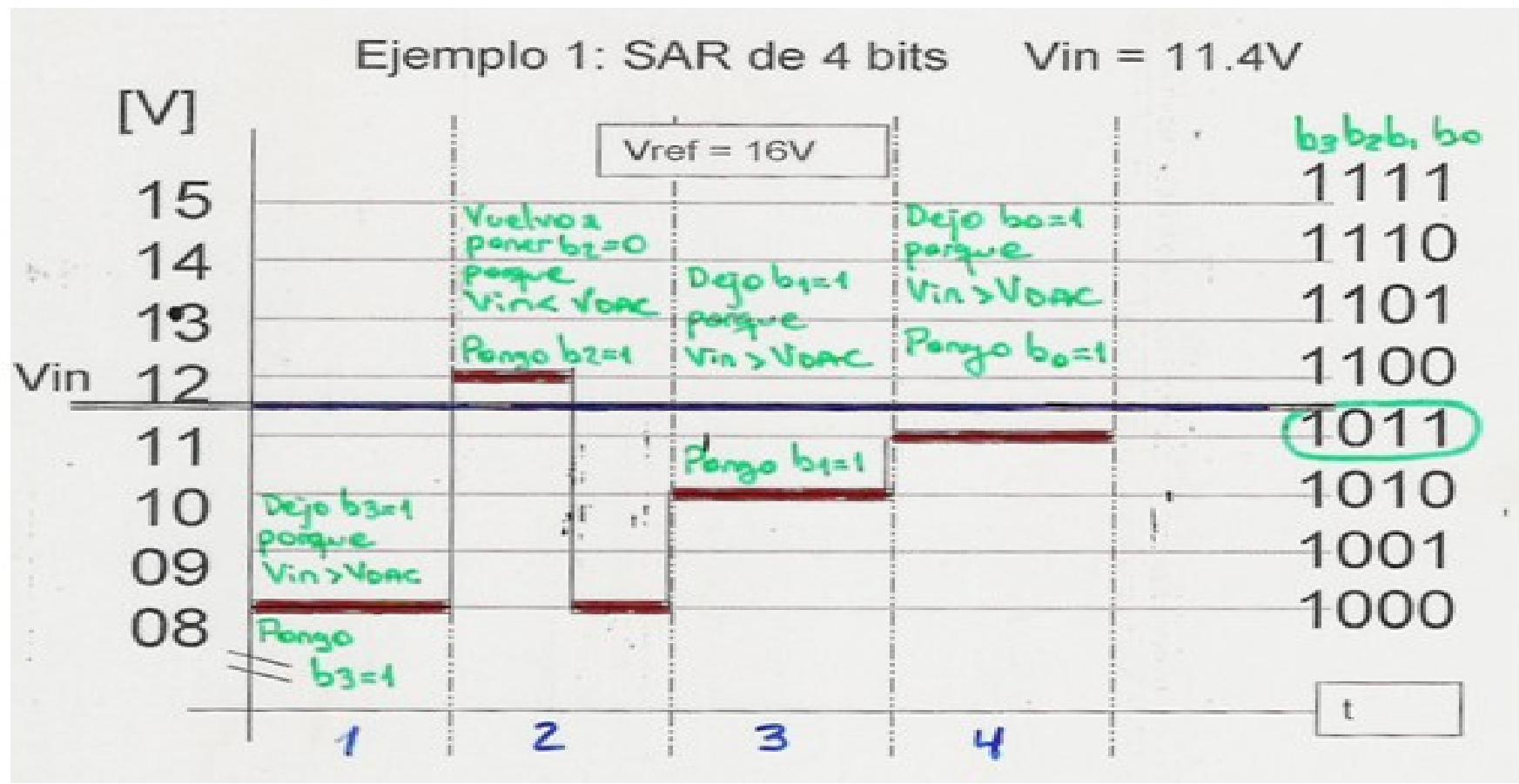
- Arquitectura idónea para resoluciones medias-altas y frecuencia de muestreo menor de 5 MS/s
- Resolución de 8 a 16 bits
- Bajo consumo

ADC DE APROXIMACIONES SUCESIVAS

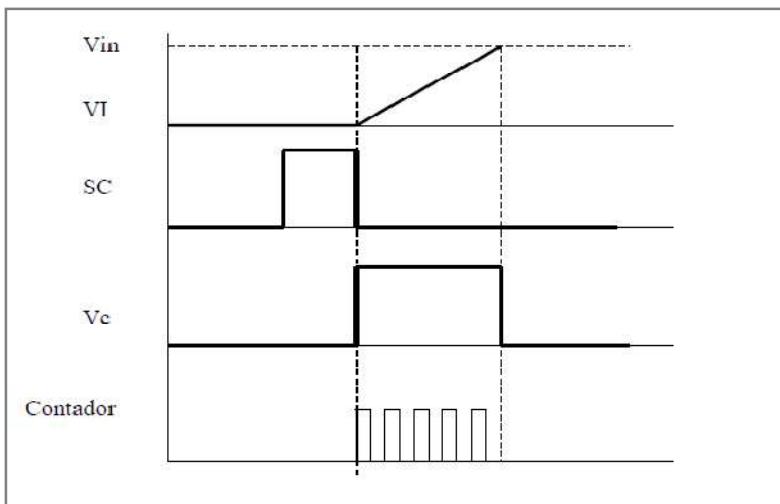
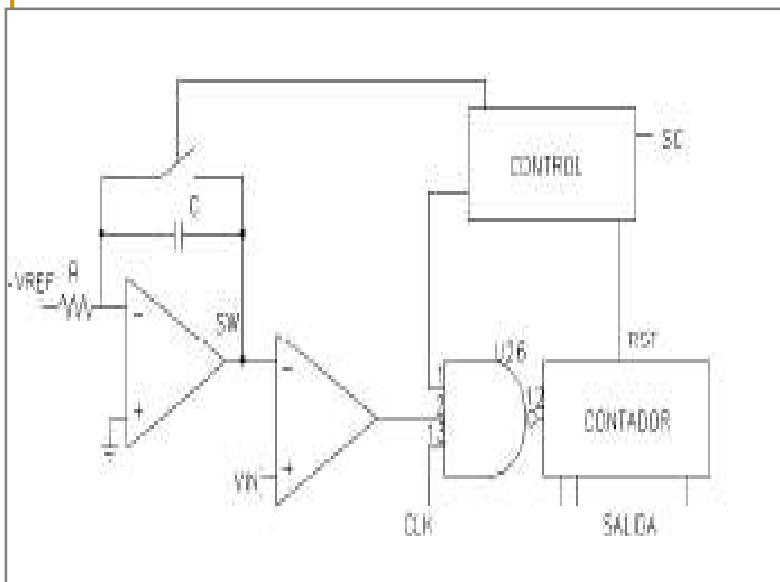


- El *ADC SAR* opera utilizando un algoritmo binario de búsqueda.
- La combinación binaria de salida se obtiene después de N comparaciones entre V_{in} y V_{DAC}
- La velocidad de conversión del *ADC SAR* viene limitada por el tiempo de establecimiento del *DAC* y la velocidad de respuesta del comparador

ADC DE APROXIMACIONES SUCESIVA



ADC DE SIMPLE RAMPA



- Proporcionan alta resolución en la conversión
- Alto rechazo al ruido de red (50/60Hz)
- Muy baja velocidad de muestreo (100S/s)
- Necesita componentes externos (R y C)
- Aplicaciones en multímetros digitales

Simple Rampa:

- Se integra la V_{IN} hasta que la salida del integrador es igual a la de referencia
- El tiempo T_{INT} transcurrido se mide con un contador
- La precisión depende de la frecuencia de reloj, estabilidad de Vref y de la tolerancia de R y C.

Lo que se ha hecho ha sido convertir V_{IN} en una magnitud de tiempo t y ese tiempo lo evaluamos con un contador...

$$t = \text{nº de pulsos}/f_{clk} = n T_{clk}$$

En el integrador tenemos:

$$V_I = -\frac{(-V_{REF})}{RC} t = \frac{V_{REF}}{RC} t$$

En el instante de tiempo $t=t_1 \Rightarrow V_I = V_{IN}$, sustituyendo...

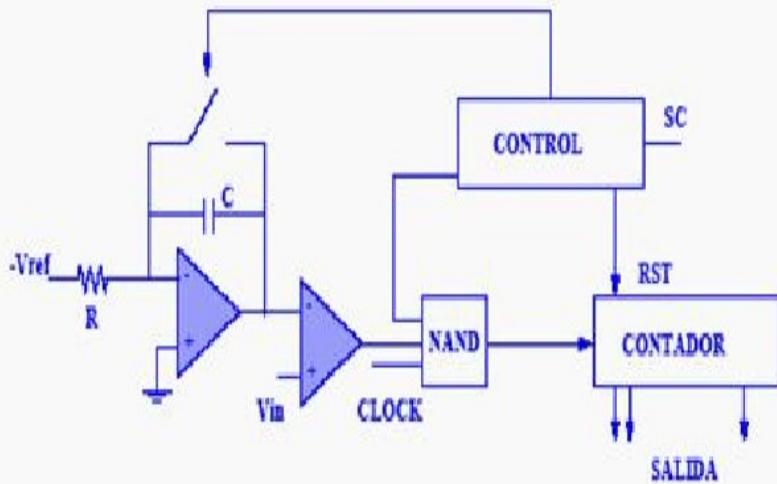
$$V_{IN} = \frac{V_{REF}}{RC} t_1 = \frac{V_{REF} n T}{RC}$$

despejando el número de pulsos n...

$$n = \frac{RC}{V_{REF} T} V_{IN}$$

ADC RAMPA SIMPLE

A/D: Convertidores de Rampa Simple



$$\text{numero de pulsos} = \frac{RC}{V_{Ref} \cdot T} \cdot V_{in}$$

Lo que se ha hecho ha sido convertir V_{IN} en una magnitud de tiempo t y ese tiempo lo evaluamos con un contador...

$$t = n^{\circ} \text{ de pulsos} / f_{clk} = n T_{clk}$$

En el integrador tenemos:

$$V_I = -\frac{(-V_{REF})}{RC} t = \frac{V_{REF}}{RC} t$$

En el instante de tiempo $t=t_1 \Rightarrow V_I = V_{IN}$, sustituyendo...

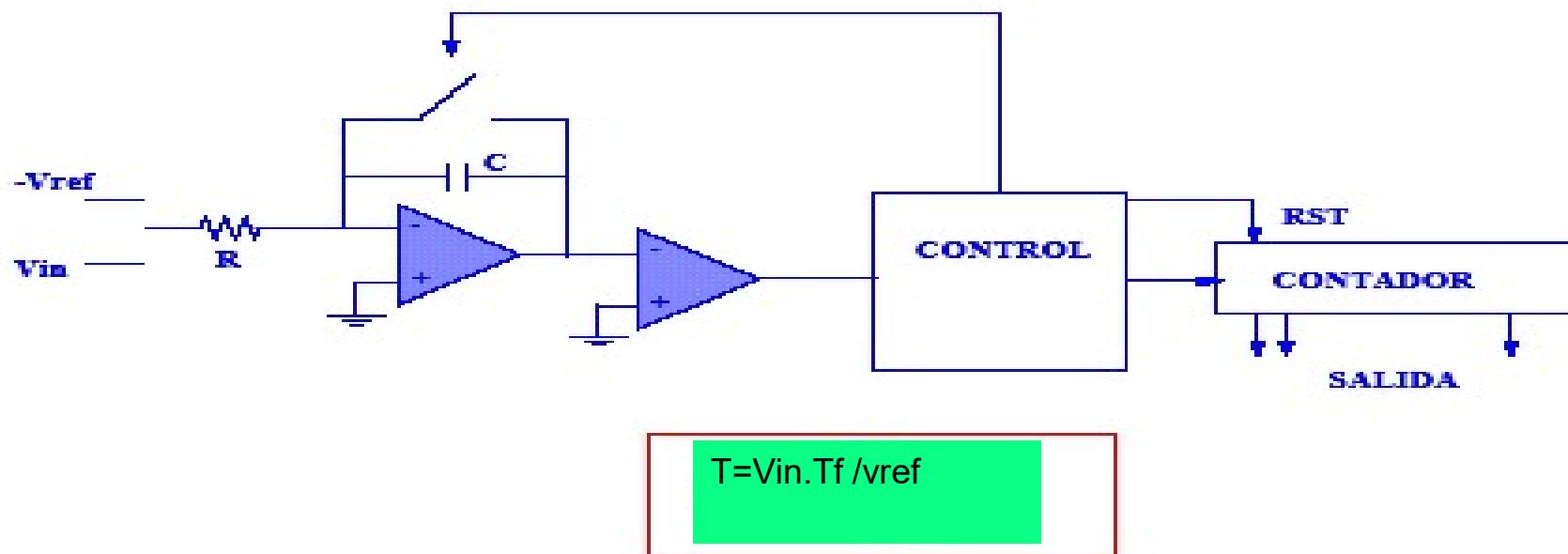
$$V_{IN} = \frac{V_{REF}}{RC} t_1 = \frac{V_{REF} n T}{RC}$$

despejando el número de pulsos n ...

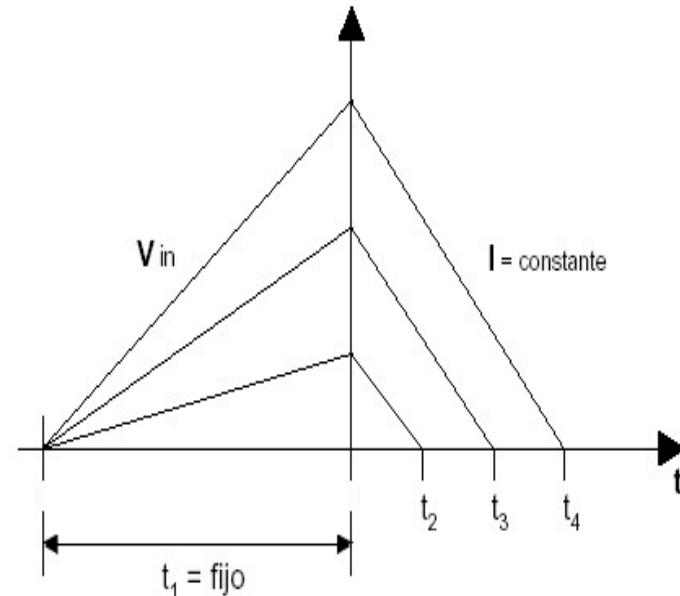
$$n = \frac{RC}{V_{REF} T} V_{IN}$$

ADC DE DOBLE RAMPA

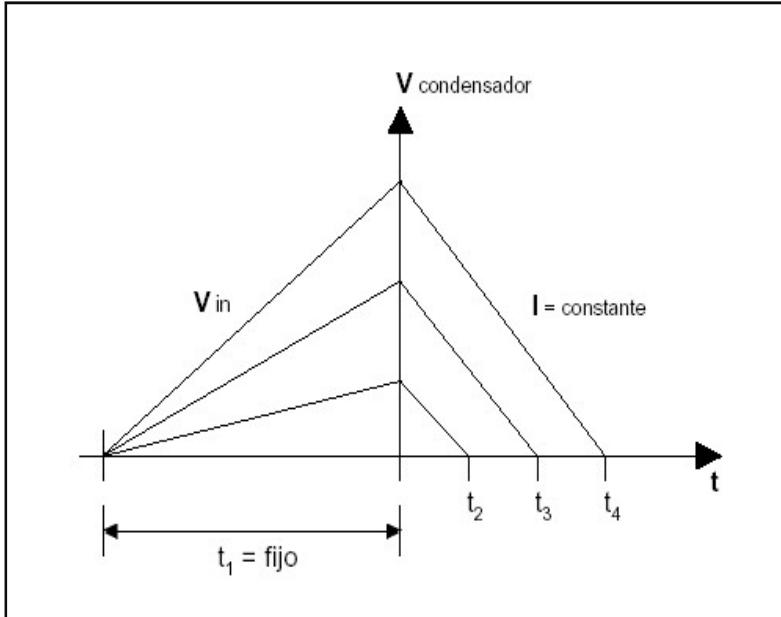
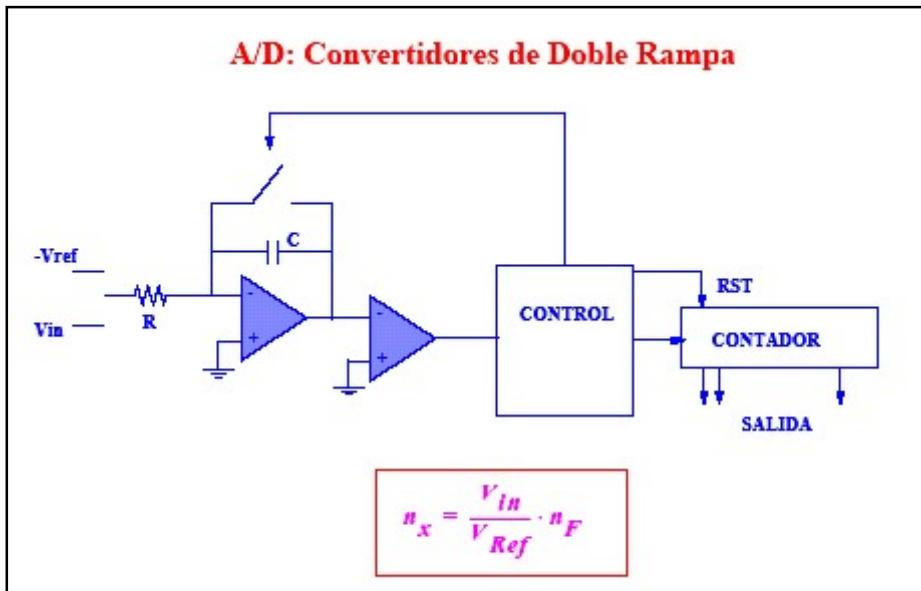
A/D: Convertidores de Doble Rampa



El ADC de doble rampa usa un condensador conectado a un voltaje de referencia. El voltaje del condensador empieza a cero y es cargado durante el tiempo de establecimiento t_1 por el voltaje de salida de un bloque muestreo y retención, V_{in} . El condensador es entonces puesto a un voltaje negativo conocido y cargado en la dirección opuesta (descargado) hasta que alcance 0 voltios otra vez. El tiempo tomado para este segundo ciclo de carga se mide usando un contador digital. Con el contador puesto inicialmente a 0, el valor final del contador da una salida digital proporcional a V_{in} .



ADC DE DOBLE RAMPA



Doble Rampa:

- Se integra la V_{IN} durante un tiempo fijo y a continuación se descarga el condensador con una corriente conocida
- El tiempo T_{DE-INT} es proporcional a la V_{IN} y no depende de R y C ni de la frecuencia de reloj
- La conversión es robusta a la tolerancia de componentes

V_{INT}

$$V_I = -\frac{V_A}{RC}t \quad V_I = -\frac{(-V_{REF})}{RC}t = \frac{V_{REF}}{RC}t$$

ADC DE DOBLE RAMPA

Durante el primer período de tiempo la salida será: $V_I = -\frac{V_A}{RC}t$

En el segundo tramo, al conmutar la entrada ésta se hace negativa lo que implica una pendiente positiva. Sin considerar las condiciones iniciales la salida sería:

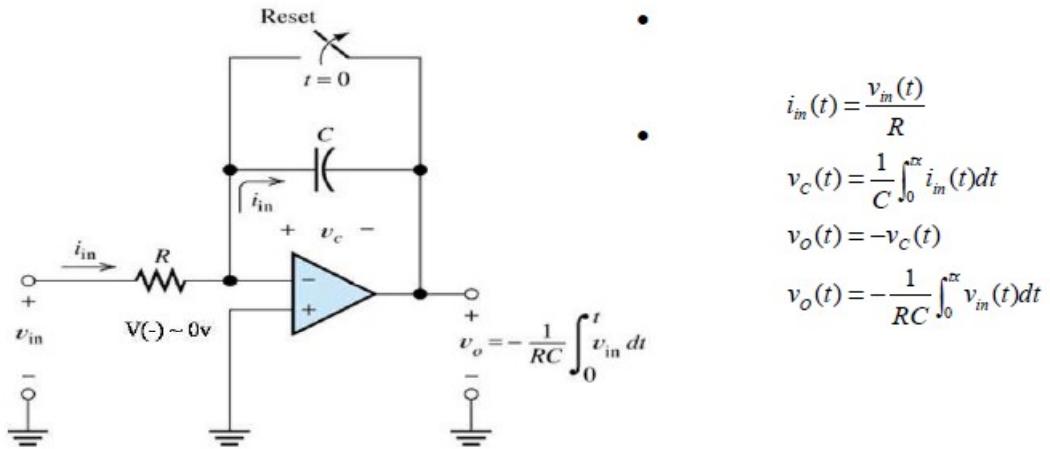
$$V_I = -\frac{(-V_{REF})}{RC}t = \frac{V_{REF}}{RC}t$$

La condición de final de segunda rampa se tendrá cuando la salida sea nula.

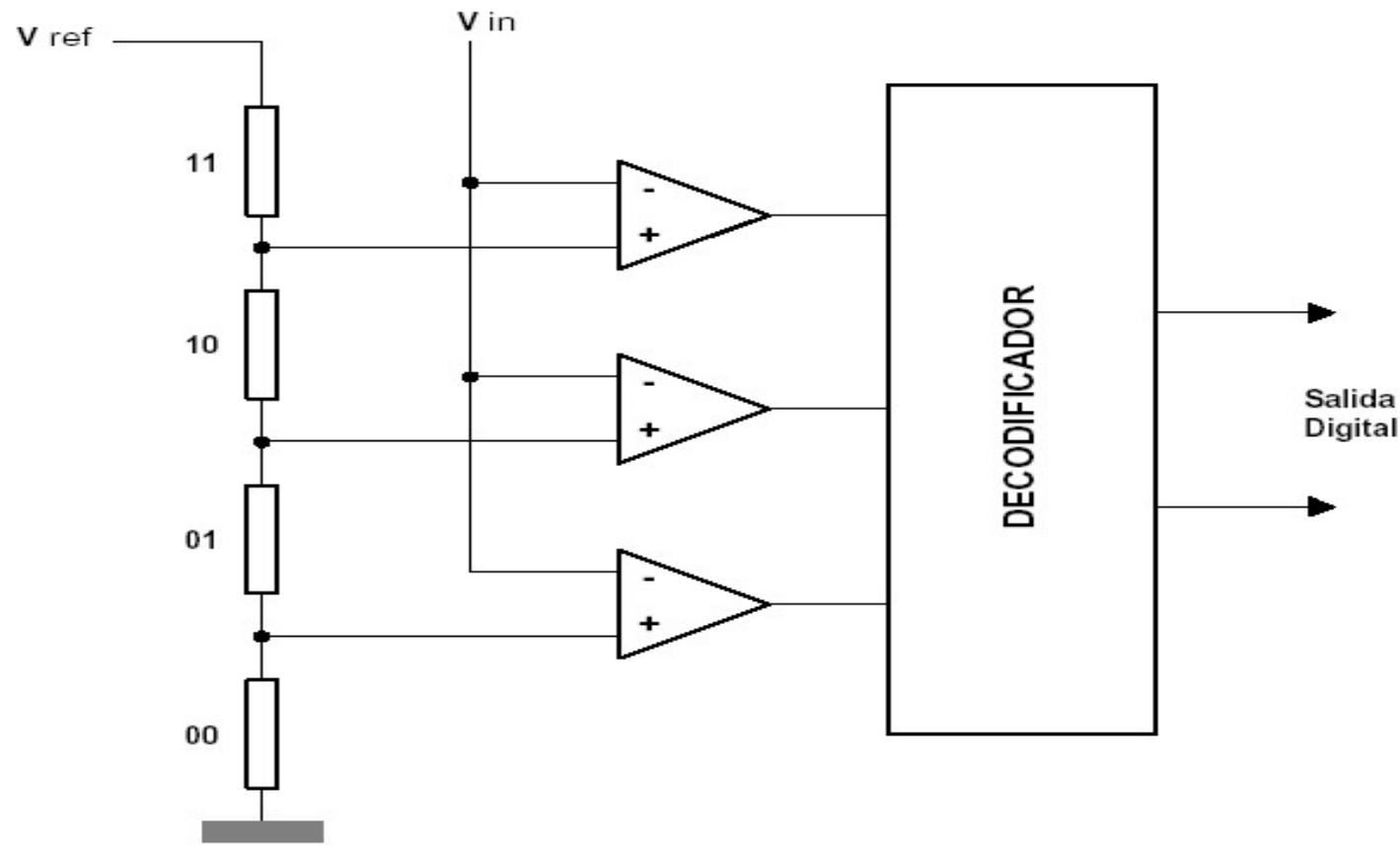
$$0 = -\frac{V_A}{RC}t_F - \frac{V_{REF}}{RC}t \Rightarrow t = \frac{V_A}{V_{REF}}t_F = t_x$$

$$\begin{aligned} i_m(t) &= \frac{v_m(t)}{R} \\ v_c(t) &= \frac{1}{C} \int_0^{\infty} i_m(t) dt \\ v_o(t) &= -v_c(t) \\ v_o(t) &= -\frac{1}{RC} \int_0^{\infty} v_m(t) dt \end{aligned}$$

Cálculos auxiliares



ADC'S INSTANTÁNEOS (FLASH)



Los ADC'S flash paralelos convierten el voltaje analógico de entrada más rápidamente que

los otros tipos de ADC's. La figura 3.31 ilustra el principio con un ADC de 2 bits. El voltaje de entrada,

V_{in} es comparado con un conjunto de voltajes de referencia los cuales aparecen al aplicar una tensión de referencia V_{ref} a una escala de resistencias de igual valor, de modo que para un ADC de n bits

necesitaremos 2^n resistencias. Los voltajes son comparados en los comparadores cuya salida es

decodificada. Las resistencias deben, en teoría, ser exactamente iguales lo cual en la práctica es

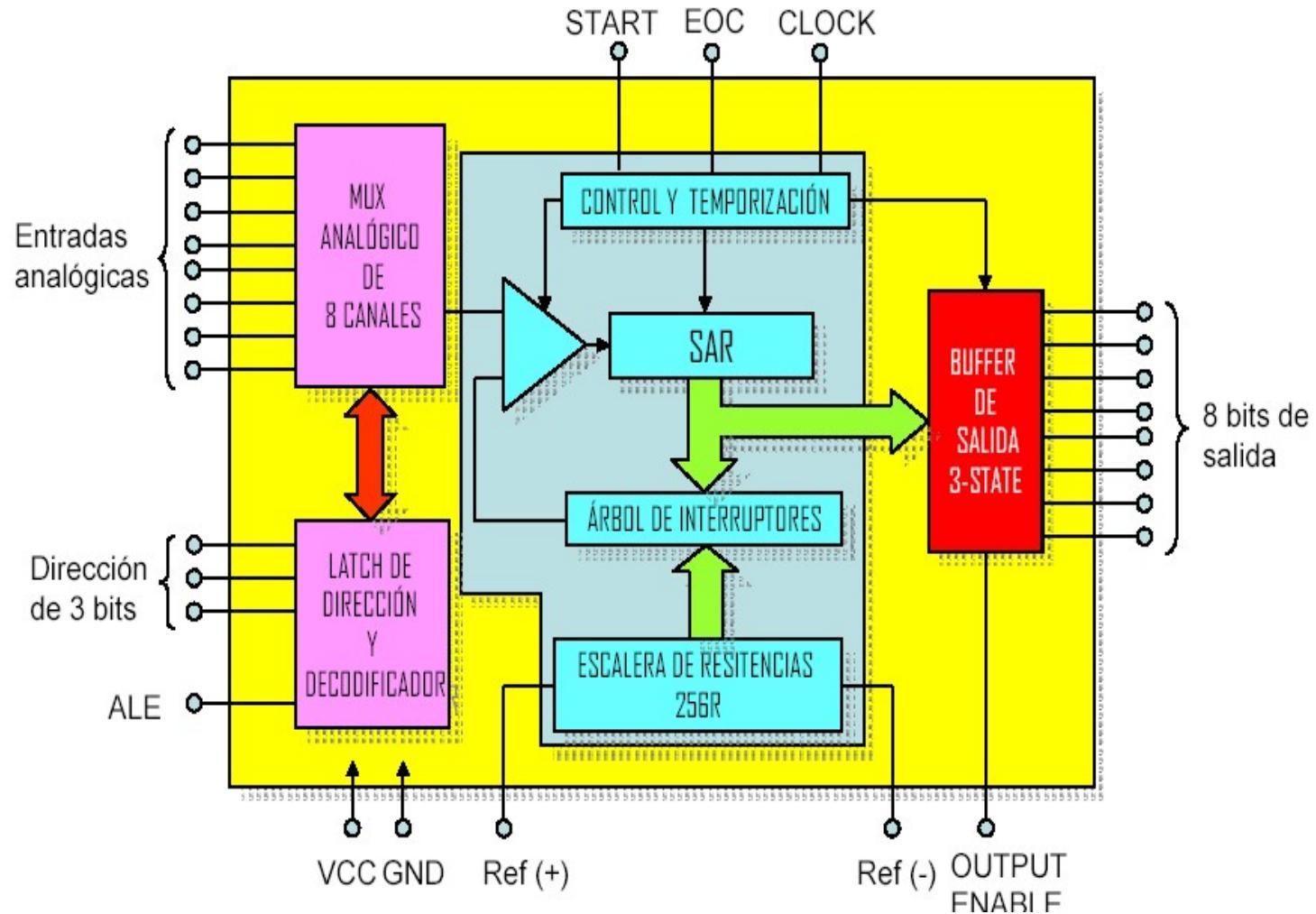
imposible ya que no podemos encontrar dos resistencias iguales. Esto hace que los ADC's flash

comerciales no suelan pasar de 8 bits (necesitamos 256 resistencias iguales).

CARACTERÍSTICAS DEL ADC0809

- Conversión por aproximaciones sucesivas
- Tiempo de conversión $100\mu s$
- 8 canales multiplexados con lógica de control de latch
- Fuente de poder de 5v DC
- Resolución de 8 bits
- Rango de voltajes de entrada de 0 a 5v con fuente de 5 volts única
- bajo consumo de potencia 15 mW
- Tecnología CMOS
- Entradas Analógicas de una terminal

DIAGRAMA DE BLOQUES DEL CONVERTIDOR ANÁLOGO-DIGITAL MULTIPLEXADO DE 8 CANALES ADC0809



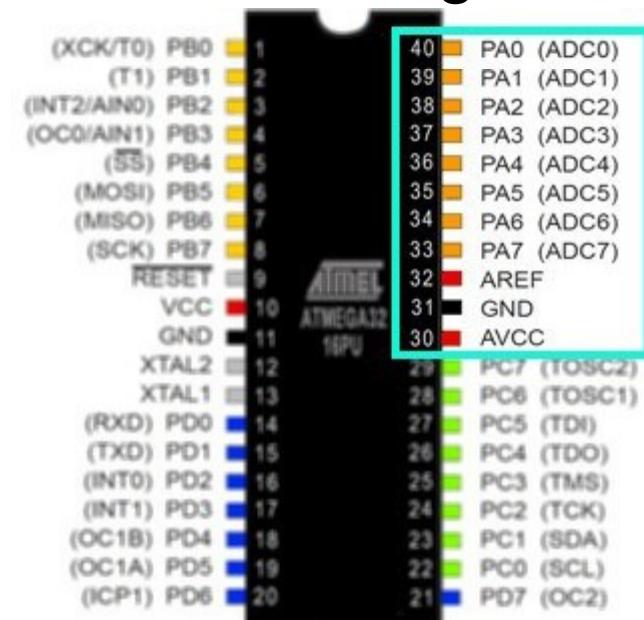
CONVERSOR DE ANALÓGICO A DIGITAL

CONFIGURACIÓN Y REGISTROS

ATmega164P:

ADC in AVR

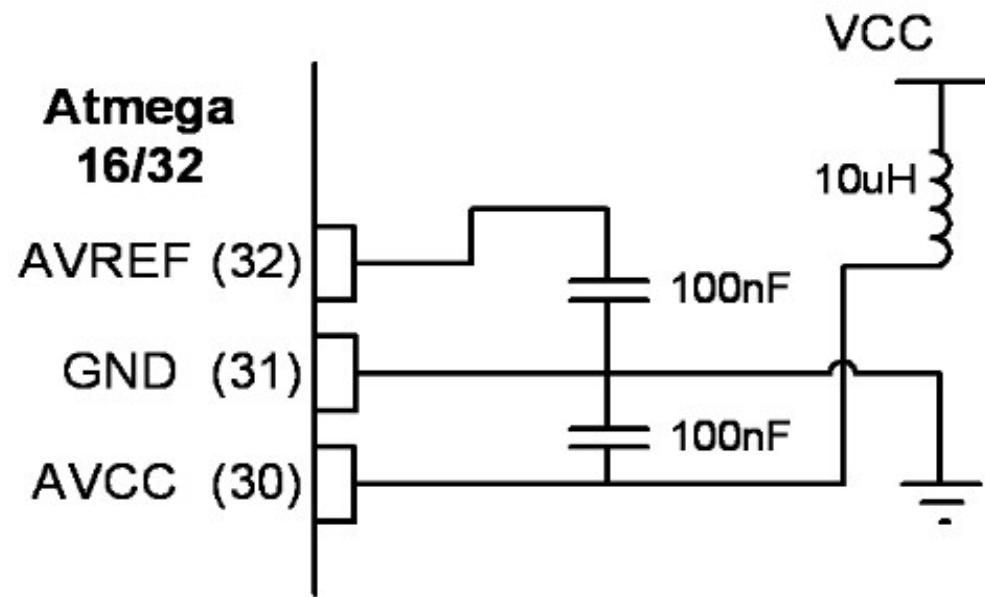
- Atmega 16/32 have internal ADC
 - 8 analogue input channel
 - 7 differential input channel
 - 2 differential input channel with 10x or 200x gain
 - 3 source of Vref
 - Internal 2.56V Vref generator



ADCs en los AVR

- El ATMega328 incluye un ADC de aproximaciones sucesivas de 10 bits, el resultado de la conversión queda en dos registros I/O: ADCH y ADCL (en Lenguaje C el par puede ser tratado como ADCW).
- El ADC está conectado a un multiplexor analógico que permite seleccionar 1 de 8 canales externos (ADCo, ADC1, ADC2,). Aunque el ATMega328 con encapsulado PDIP sólo tiene 6 canales.
- Para alcanzar la máxima resolución, el ADC debe trabajar con una frecuencia entre 50 y 200 kHz, la cual se puede generar con un pre-escalador, partiendo de la frecuencia base del microcontrolador.
- La primera conversión requiere inicializar la circuitería analógica, por lo que invierte 25 ciclos de reloj. Las siguientes conversiones solo utilizan 13 ciclos de reloj.

Hardware Consideration



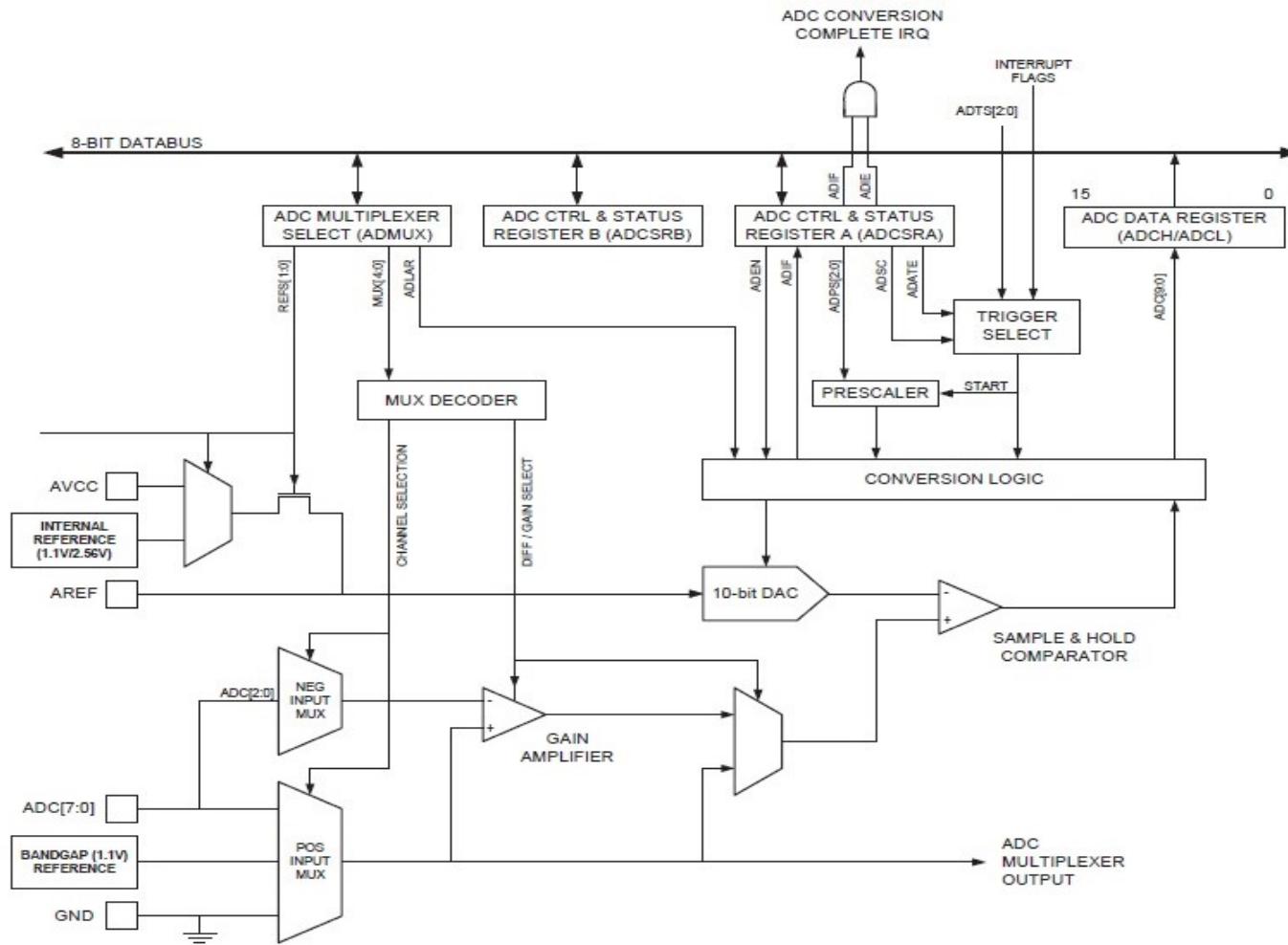
CARACTERÍSTICAS DEL ADC

- 1. 10 BITS DE RESOLUCIÓN, POR APROXIMACIONES SUCESIVAS**
- 2. VOLTAJE DE ENTRADA DE 0 A V_{cc}**
- 3. MODOS DE CONVERSIÓN LIBRE O UNA SOLA conversión**
- 4. VOLTAJE DE REFERENCIA SELECCIONABLE DE 2,56 V o DE 1,1 V**

CARACTERÍSTICAS DEL ADC

- 1. DISPARO AUTOMÁTICO PARA EL INICIO DE CONVERSIÓN CON ALGUNAS INTERRUPCIONES**
- 2. TIEMPO DE CONVERSIÓN DE 13 A 260 uS**

DIAGRAMA DE BLOQUES DEL ADC



Registros

1. **ADCSRA** → REGISTRO A DE CONTROL Y DE ESTADO DEL ADC
 2. **ADCSR B** → REGISTRO B DE CONTROL Y DE ESTADO DEL ADC
 3. **ADMUX** → REGISTRO DE SELECCIÓN DEL MULTIPLEXER DEL ADC
 4. **ADCH** → REGISTRO RESULTADO CONVERSIÓN H
 5. **ADCL** → REGISTRO RESULTADO CONVERSIÓN L
-
1. **DIRO** → REGISTRO PARA DESHABILITAR LAS ENTRADAS DEL PUERTO

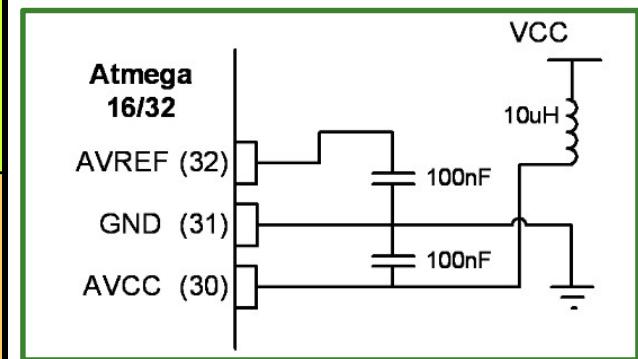
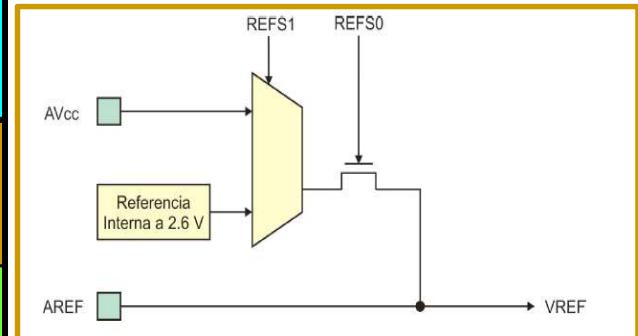
REGISTRO DE SELECCIÓN DEL MULTIPLEXER DEL ADC

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
ADMUX	REFS1	REFS0	ADLAR	MUX2	MUX3	MUX2	MUX1	MUX0		(\$7C)

DESCRIPCIÓN	
REFS1/0	Bits para seleccionar el voltaje de referencia
ADLAR	Ajuste del resultado a la izquierda
MUX4..0	Bits de selección del canal analógico y la ganancia

VOLTAJE DE REFERENCIA PARA EL ADC

REFS1/0	DESCRIPCIÓN
00	Voltaje de referencia externa en el pin AREF, el interno está desconectado
01	Voltaje de referencia externo en el pin AVCC, con un condensador en el pin AREF
10	Voltaje de referencia interno de 1,1 voltios, con un condensador en el pin AREF
11	Voltaje de referencia interno de 2,56 voltios, con un condensador en el pin AREF



REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
ADMUX	REFS1	REFS0	ADLAR	MUX2	MUX3	MUX2	MUX1	MUX0		(\$7C)

AJUSTE DEL RESULTADO

ADLAR

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
ADCH	-	-	-	-	-	-	ADC9	ADC8		(\$79)
ADCL	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0		(\$78)

ADLAR=0

ADLAR=1

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
ADCH	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2		(\$79)
ADCL	ADC1	ADC0	-	-	-	-	-	-		(\$78)

PRIMERO LEER ADCL, LUEGO ADCH

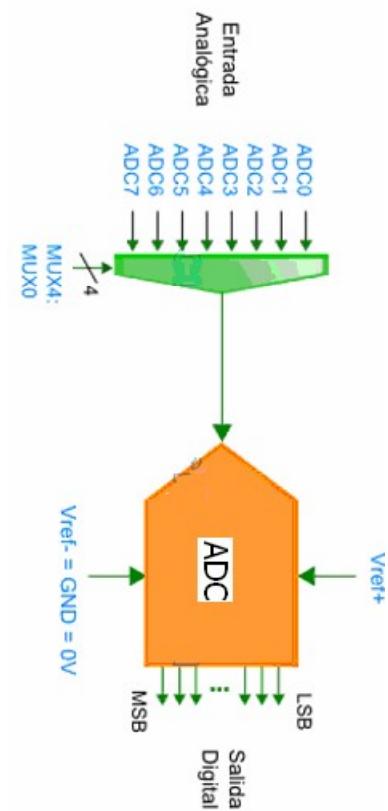
REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
ADMUX	REFS1	REFS0	ADLAR	MUX2	MUX3	MUX2	MUX1	MUX0		(\$7C)

ENTRADA DE UN SOLO CANAL

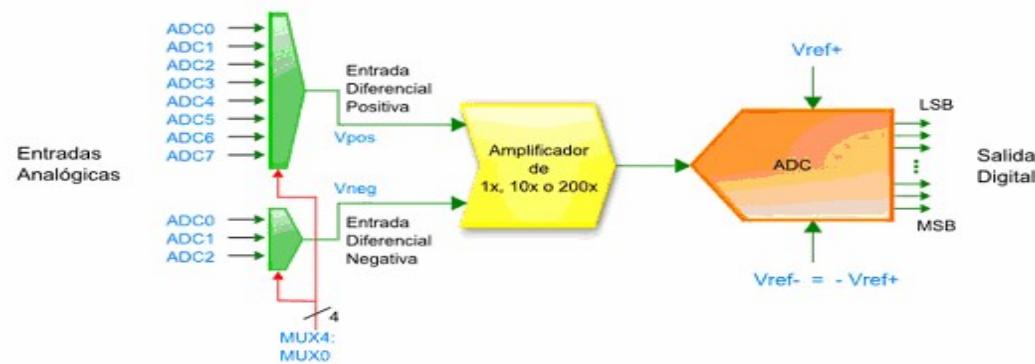
MUX4:0

MUX4/3/2/1/ 0	CANAL SELECCIONADO
00000	Pin ADC0 = PA0
00001	Pin ADC1 = PA1
00010	Pin ADC2 = PA2
00011	Pin ADC3 = PA3
00100	Pin ADC4 = PA4
00101	Pin ADC5 = PA5
00110	Pin ADC6 = PA6
00111	Pin ADC7 = PA7

Modo de entrada única



REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
ADMUX	REFS1	REFS0	ADLAR	MUX2	MUX3	MUX2	MUX1	MUX0		(\$7C)



MUX4/3/2/1/0	ENTRADA POSITIVA	ENTRADA NEGATIVA	GANANCIA
10000	ADC0	ADC1	1x
10001	ADC1	ADC1	1x
10010	ADC2	ADC1	1x
10011	ADC3	ADC1	1x
10100	ADC4	ADC1	1X
10101	ADC5	ADC1	1X
10110	ADC6	ADC1	1X
10111	ADC7	ADC1	1X

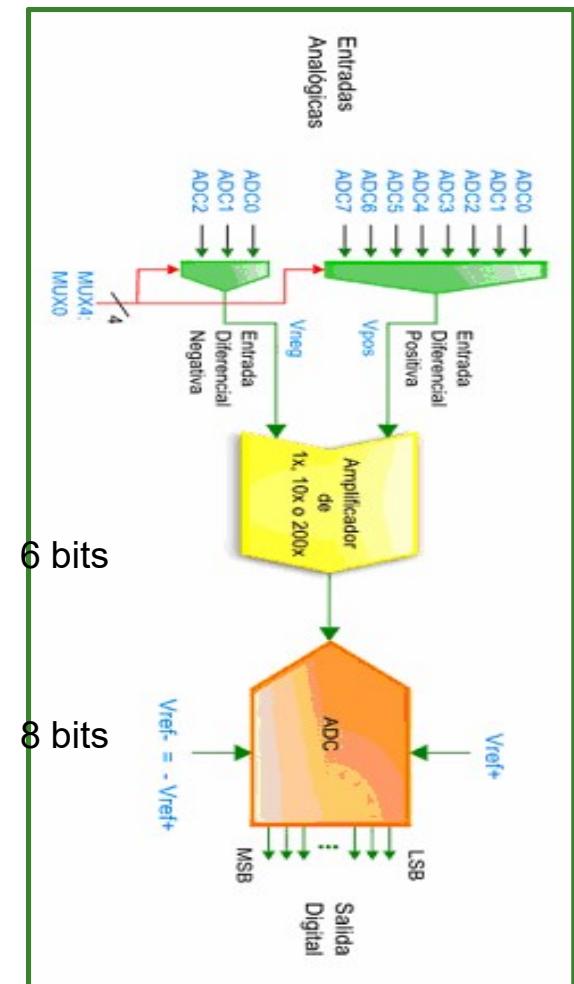
ENTRADA DIFERENCIAL CON ADC1 DE REFERENCIA

ENTRADA DIFERENCIAL CON ADC2 DE REFERENCIA

MUX4/3/2/1/0	ENTRADA POSITIVA	ENTRADA NEGATIVA	GANANCIA
11000	ADC0	ADC2	1x
11001	ADC1	ADC2	1x
11010	ADC2	ADC2	1x
11011	ADC3	ADC2	1x
11100	ADC4	ADC2	1X
11101	ADC5	ADC2	1X
11110	$V_{bg} = 1.1 \text{ v}$		
11111	ENTRADA DE UN SOLO CANAL = 0 voltios		

ENTRADA DIFERENCIAL

MUX4/3/2/1/0	ENTRADA POSITIVA	ENTRADA NEGATIVA	GANANCIA
01000	ADC0	ADC0	10x
01001	ADC1	ADC0	10x
01010	ADC0	ADC0	200x
01011	ADC1	ADC0	200x
01100	ADC2	ADC2	10X
01101	ADC3	ADC2	10X
01110	ADC2	ADC2	200X
01111	ADC3	ADC2	200X



REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
ADMUX	REFS1	REFS0	ADLAR	MUX2	MUX3	MUX2	MUX1	MUX0		(\$7C)

RESULTADO DE LA CONVERSIÓN

Entrada de un solo canal

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

Entrada diferencial

$$ADC = \frac{(V_{POS} - V_{NEG}) \cdot GAIN \cdot 512}{V_{REF}}$$

REGISTRO A DE CONTROL Y DE ESTADO DEL ADC

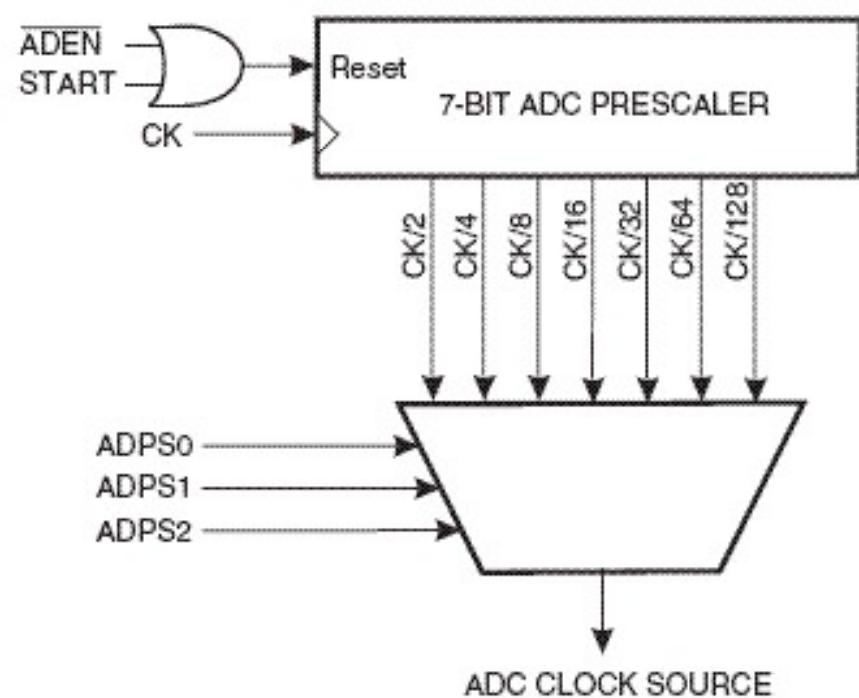
REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0		(\$7A)

DESCRIPCIÓN	
ADEN	Para habilitación del ADC
ADSC	Bit de inicio de la conversión
ADATE	Para habilitar el disparo automático
ADIF	Bandera de Interrupción de conversión completa
ADIE	Para habilitar la interrupción de conversión
ADPS2/1/0	Selección de la velocidad de conversión

RELOJ DEL CONVERSOR

ADPS2/1/0	DIVISOR
000	2
001	2
010	4
011	8
100	16
101	32
110	64
111	128

Con 10 bits de resolución adc clock entre 50KHz y 200KHz
 Menor resolución $\geq 200\text{KHz}$



Ejemplos

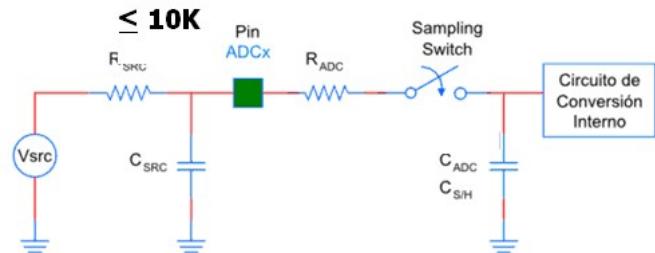
Con 10 bits de resolución adc clock entre 50KHz y 200KHz

Menor resolución $\geq 200\text{KHz}$

- $F_{clock}=8 \text{ MHZ}$
- SI ADPS2:ADPS0=111 $\rightarrow F_{CADC} = 8\text{MHz}/128=62.5\text{KHz} \rightarrow \text{OK}$
- SI ADPS2:ADPS0=101 $\rightarrow F_{CADC} = 8\text{KHz}/32=250\text{KHz} \rightarrow \text{BAD}$
(solo los 8 bit mas significativos)

TABLA TIPO DE CONVERSIÓN

Tipo de Conversión	Muestreo y retención (Ciclos desde el inicio de la conversión)	Tiempo de Conversión (Ciclos)
Primera conversión	14.5	25
Conversiones Normales, de entrada única	1.5	13
Conversiones Normales, de entrada diferencial	1.5/2.5	13/14
Conversiones auto-disparadas	2	13.5



Entrada única $\rightarrow 13+1.5 = 14.5$ ciclos. Si el ADC trabajara a su máxima frecuencia recomendada de 200kHz , cada ciclo duraría $1/200\text{kHz} = 5\text{us}$. cada una de estas conversiones en $14.5*5 = 72.5\text{us}$

Consideraciones

$$\text{Resolución} = \frac{V_{ref+} - V_{ref-}}{2^n - 1}$$

$$\text{Valor ADC} = \frac{2^n - 1}{V_{ref+}} \times V_{in}$$

$$V_{in} = \text{Valor ADC} \times \frac{V_{ref+}}{2^n - 1}$$

$$V_{in} = \text{Valor ADC} \times \text{Resolución}$$

$$F_{ADC} = \frac{\text{Frecuencia CPU}}{\text{Preescaler}}$$
$$F_{muestreo} = \frac{F_{ADC}}{13} \quad (25 \text{ primera conversión})$$
$$T_{muestreo} = \frac{1}{F_{muestreo}}$$

Suponiendo que nuestro micro trabaje a 8MHz y un prescaler de 64, tendríamos:

$$F_{ADC} = 8000000/64 = 125000 \text{ (125KHz)},$$

$$F_{muestreo} = 125000/13 = 9615.38 \text{ (9.6KHz)}$$

y el periodo de muestreo:

$$T_{muestreo} = 1/9615 = 104\text{us.}$$

Esto quiere decir que una conversión del ADC a una frecuencia de 8MHz y un prescaler de 64 tardaría un tiempo de 104 us. Es posible utilizar frecuencias para el ADC mayores a 200KHz, pero la resolución estable baja a 8 bits.

REGISTRO B DE CONTROL Y DE ESTADO DEL ADC

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
ADCSRB	-	ACME	-	-	-	ADTS2	ADTS1	ADTS0		(\$7B)

ADTS2/1/0	DESCRIPCIÓN
000	Modo de libre conversión
001	Por el comparador analógico
010	Por pedido de Interrupción Externa 0
011	Por emparejamiento del Comparador del Timer 0
100	Por desbordamiento del Timer 0
101	Por emparejamiento del Comparador B del Timer 1
110	Por desbordamiento del Timer 1
111	Por la captura del Timer 1

REGISTRO PARA DESHABILITAR LAS ENTRADAS DEL PUERTO

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
DIDR0	ADC7D	ADC6D	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D		(\$7E)

En estos bits se pone uno, para deshabilitar el buffer de la entrada digital del respectivo Pórtico; lo que permite una reducción en el consumo de potencia

Secuencia simplificada de configuración de un adc

- 1.- Desabilite interrupciones. Recomendado pero no necesario.
- 2.- Configure como entrada los pines del microcontrolador que utilizará como entradas analógicas.
- 3.- Seleccione la frecuencia de la señal AD Clock. Ver Tabla ADC-5.

;PASO1

cli

;SREG<I>=0.int. desabilitadas.

;PASO2

ldi r17,0x00

out DDRA,r17

;Configura como entrada Puerto A

;Son las entradas Analógicas

;PASO3

cbi ADCSRA,ADPS2

sbi ADCSRA,ADPS1

cbi ADCSRA,ADPS0

;Frecuencia de la señal ADC Clock

;es fosc/4

;ADPS0=0 ADPS1=1 ADPS2=0

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0		(\$7A)

4.- Seleccione el modo de operación:

Modo de una Sola Conversión ADCSR<ADFR>="0"

Modo de Conversión Continua ADCSR<ADFR>="1"

5.- Si lo requiere, habilite la interrupción del ADC. Por lo general sí se habilita la interrupción.

ADCSR<ADIE>="1"

6.- Limpie posible interrupción falsa del ADC, para esto escriba un "1" en el bit ADIF del registro ADCSR.

;PASO4

sbi ADCSRA,ADFR

;Modo Conversión Continua

;PASO5

sbi ADCSRA,ADIE

;Habilita Int ADC

;PASO6

sbi ADCSRA,ADIF

;Limpia Posible Interrupción Falsa

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0		(\$7A)

7.- Seleccione el canal de entrada analógico deseado

8.- Habilite el ADC.

9.- Habilite interrupciones globales.

10.- Inicie conversión. $\text{ADCSR} < \text{ADSC} > = 1$ "

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
ADMUX	REFS1	REFS0	ADLAR	MUX2	MUX3	MUX2	MUX1	MUX0		(\$7C)

;PASO7

cbi ADMUX,MUX2 ;Selecciona Canal 0

cbi ADMUX,MUX1 ;Pin 40, PA0

cbi ADMUX,MUX0

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0		(\$7A)

;PASO8

sbi ADCSRA,ADEN ;Habilita ADC

;PASO9

sei ;Habilita Interrupciones Globales

;PASO10

sbi ADCSRA,ADSC ;Inicia la primera conversión.

```
.org 0x0000  
rjmp reset
```

; reset vector
; jump to "reset"

```
.org 0x000E  
rjmp ADC_ISR
```

; ADC Conversion Complete ;Interrupt vector:
; jump the "ADC_ISR"

```
reset:  
ldi r16, low(RAMEND)  
out SPL, r16  
ldi r16,  
high(RAMEND)  
out SPH, r16
```

; the reset code:

```
; stack setup; set SPH:SPL to  
; RAMEND
```

```
ldi r16, 0xFF  
out DDRD, r16
```

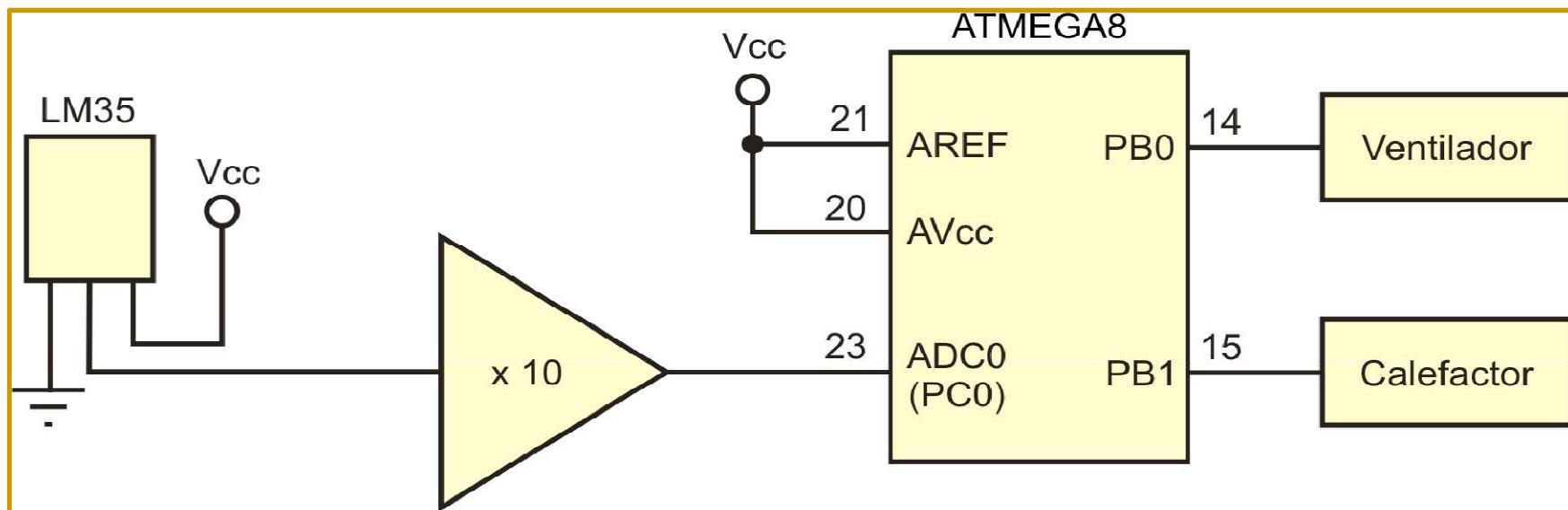
; set all PortD pins to output

ldi r16, 0	; write zero
out ADMUX, r16	; to ADMUX (select channel 0)
ldi r16, 0b11101101	; from left to right: ADC Enable, Start
out ADCSR, r16	;Conversion, Free-Running Mode,
	write
	; zero to ADC Int flag, enable int,
	prescaler: 101 for XTAL/32
sei	; enable interrupts
loop:	
rjmp loop	;; and loop
	; forever

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0		(\$7A)

Desarrollar un sistema con base en un ATMega8, que mantenga la temperatura de una habitación en el rango entre 18 y 23 °C, aplicando la siguiente idea: Si la temperatura es mayor a 23 °C, el sistema debe activar un ventilador, si es menor a 18 °C, el sistema debe activar un calefactor.

Para temperaturas entre 18 y 23°C no se debe activar alguna carga.



El Sensor de temp. LM35, entrega 10 mV/°C.
 El sensor es amplificado por un factor de 10,
 Para temperaturas entre 0 y 50 °C, el sensor
 entrega voltajes entre 0 y 5 V

$$\text{Resolución} = \frac{V_{\text{MAX}}}{2^{10} - 1} = \frac{5\text{V}}{1023} = 4.8875\text{mV}$$

Temperatura	Voltaje analógico (Van)	Valor digital (Van/Resolución)
18 °C	1.8 V	1.8 V / 4.8875 mV = 368.28 ≈ 368
23 °C	2.3 V	2.3 V / 4.8875 mV = 470.58 ≈ 471

Ejemplo 1

Fckl=1 Mhz

2	4	8	16	32	64	128	Presc.
500K	250K	125K	62,5 K	31,25K	15,6K	7,8K	Fclock/Pres
X	X	0	0	X	X	X	

ENTRE
50 k Y 200k

$$F_{8m} = F_{clock}/presc * 13 = 9,6 \text{ KHz} \rightarrow f_{max} = 4,8 \text{ KHz}$$

$$F_{16m} = F_{clock}/presc * 13 = 94,8 \text{ KHz} \rightarrow f_{max} = 2,4 \text{ KHz}$$

cmaq.

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
ADMUX	REFS1	REFS0	ADLAR	MUX2	MUX3	MUX2	MUX1	MUX0		(\$7C)
REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0		(\$7A)

```

.include      "m8def.inc"
.EQU    inf = 368           ; Valor digital para 18 grados
.EQU    sup  = 471           ; Valor digital para 23 grados

CLR    R16
OUT    ADMUX, R16          ; Selecciona ADC0 y Vref en AREF
LDI    R16, 0b11000011
OUT    ADCSRA, R16          ; Hab. ADC, inicia conversión y divide entre 8

LDI    R16, 0xFF
OUT    DDRB, R16            ; Puerto B como salida
CLR    R16
OUT    PORTB, R16           ; Inicia con las salidas sin activar

LOOP: SBIS  ADCSRA, ADIF      ; Espera fin de conversión
      RJMP  LOOP
      SBI   ADCSRA, ADIF      ; Limpia bandera

```

```

; Se destinan R27 y R26 para el valor digital proporcionado por el ADC
    IN      R27, ADCH
    IN      R26, ADCL

                                ; Comparación con el límite superior
    CPI      R26, LOW(sup)      ; Compara parte baja
    LDI      R16, HIGH(sup)
    CPC      R27, R16           ; Compara parte alta, considera acarreo
    BRLO    no_sup             ; brinca si temp < sup
    BREQ    no_sup             ; o si temp = sup
    LDI      R16, 0x01          ; Activa ventilador
    OUT     PORTB, R16         ; cuando temp > sup
    RJMP   fin_comp

no_sup:
    CPI      R26, LOW(inf)      ; Comparación con el límite inferior
    LDI      R16, HIGH(inf)      ; Compara parte baja
    CPC      R27, R16           ; Compara parte alta, considera acarreo
    BRSH    no_inf              ; brinca si temp >= inf
    LDI      R16, 0x02          ; Activa calefactor
    OUT     PORTB, R16         ; cuando temp < inf
    RJMP   fin_comp

no_inf:
    LDI      R16, 0x00          ; salidas sin activar
    OUT     PORTB, R16         ; cuando temp < sup y temp > inf

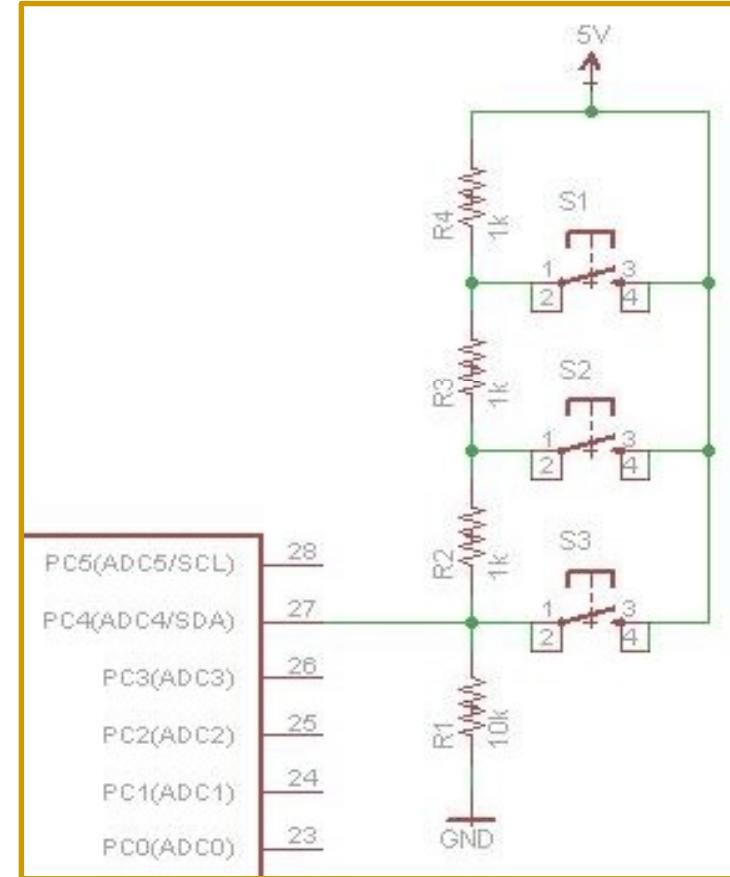
fin_comp:
    SBI      ADCSRA, ADSC      ; Inicia nueva conversión
    RJMP   LOOP                ; Regresa al lazo infinito

```

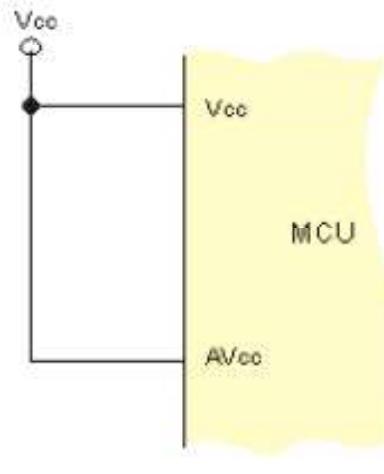
Ejemplo 2

(Uso de ADC para botones adicionales)

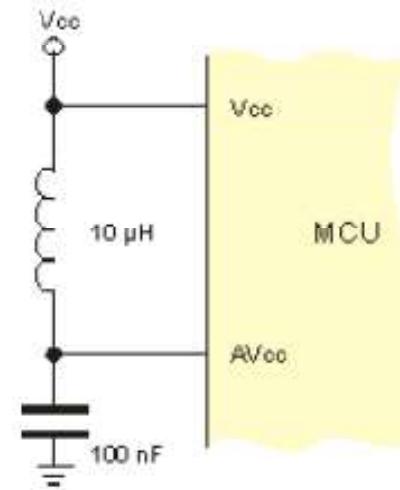
- Si no tienes suficientes entradas digitales.
- Construye divisor de voltaje y usas los botones para poner en cortocircuito las resistencias que podrías leer, usa el ADC para averiguar qué botón se presionó.
- Las desventajas de hacer esto es que toma mucho más tiempo leer un ADC que leer una entrada digital y que no puede leer varias teclas presionadas al mismo tiempo.



Conexión de AVcc



Normal



Con inmunidad al ruido

Comparador analógico (AC)

El comparador analógico es un recurso que indica la relación existente entre dos señales analógicas externas.

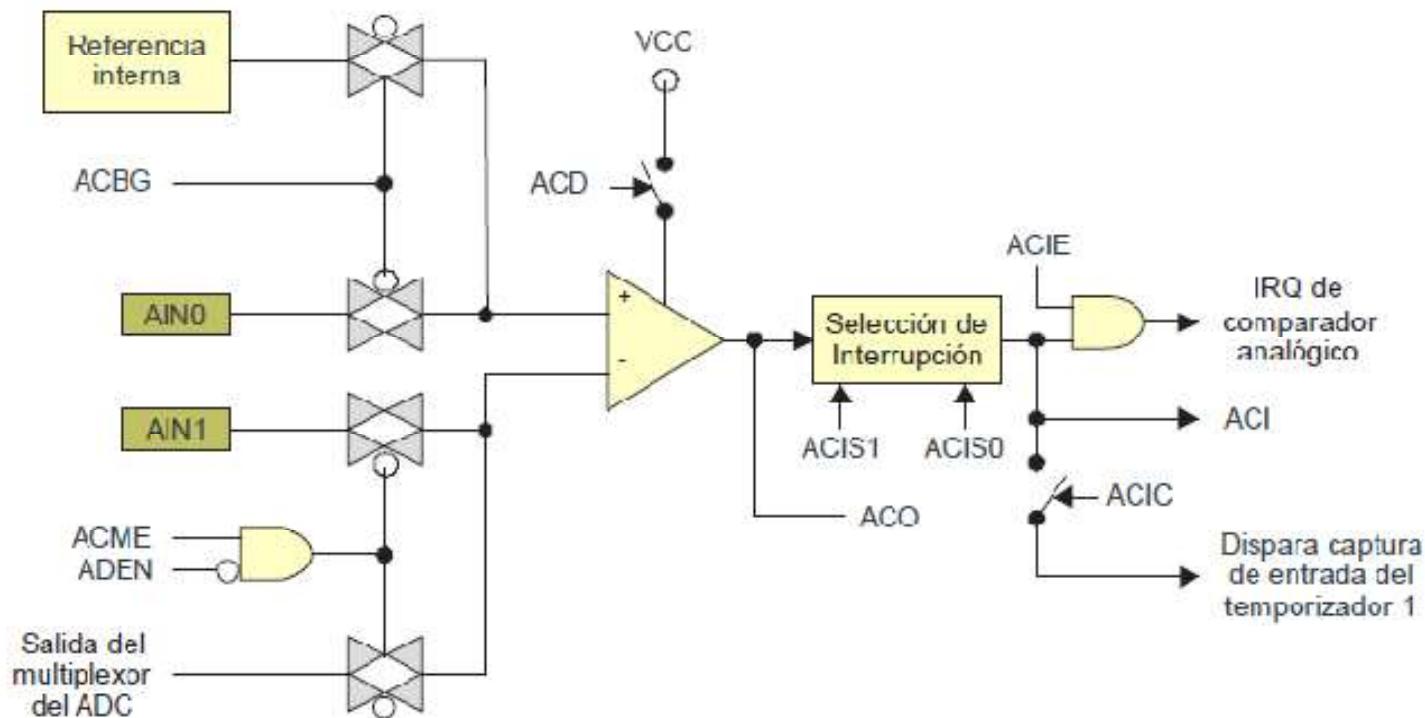
Útil para aplicaciones en donde no precisa conocer el valor digital de una señal analógica, sino que es suficiente con determinar si es mayor o menor que alguna referencia.

Se basa en un amplificador operacional en lazo abierto, el cual se va a saturar cuando una entrada analógica (**AIN0**) sea mayor que otra (**AIN1**), poniendo en alto al bit **ACO** (**ACO**, *Analog Comparator Output*).

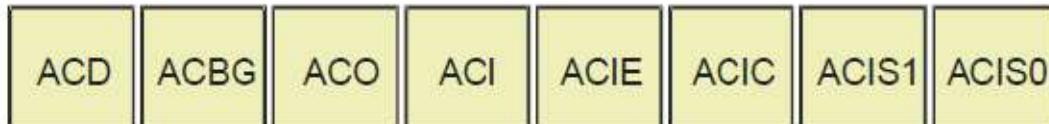
Por medio de interruptores analógicos es posible remplazar la entrada **AIN0** por una referencia interna cuyo valor típico es de 1.1 V.

La entrada **AIN1** puede remplazarse por la salida del multiplexor del ADC, con ello es posible comparar más de una señal analógica con la misma referencia. El ADC debe estar deshabilitado.

Comparador analógico (AC)



Registro ACSR. Registro Control y Estado del AC.



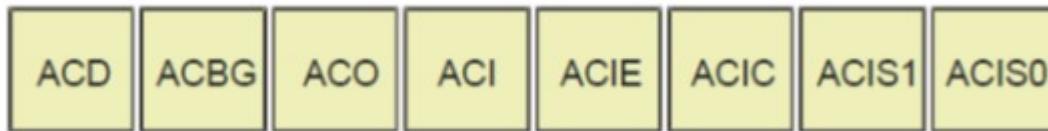
Bit 7 – ACD: AC Disable. Deshabilita el AC para minimizar el consumo de energía (con o el AC está activo). Si el AC está inactivo y se va a activar nuevamente, se sugiere deshabilitar su interrupción porque podría generar un evento.

Bit 6 – ACBG: AC Bandgap Select. Al ponerlo en alto, la entrada en la terminal positiva proviene de un voltaje de referencia interno (V_{BG}).

Bit 5 – ACO: AC Output. La salida del AC se sincroniza y se muestra en este bit, la sincronización toma 1 ó 2 ciclos de reloj.

Bit 4 – ACI: AC Interrupt Flag. Bandera para generar interrupción. Su activación está en función del resultado del comparador y de los bits **ACIS1** y **ACISO**.

ACSR.



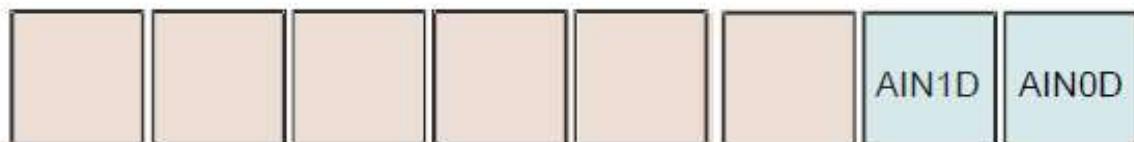
Bit 3 – ACIE: AC Interrupt Enable. Habilitador de interrupciones del comparador analógico.

Bit 2 – ACIC: AC Input Capture Enable. Habilita la función de captura de entrada del temporizador 1. La salida del comparador se conecta al hardware de captura, utilizando sus recursos de selección de flanco y cancelación de ruido. Para que el comparador dispare la interrupción por captura de entrada, el bit TCIE del registro TIMSK debe estar en alto.

Bits 1 y 0 – ACIS[1:0]: AC Interrupt Mode Select. Seleccionan el modo para la interrupción. Al modificar estos bits, se sugiere deshabilitar la interrupción por que se podría generar eventos no deseados.

ACIS1	ACIS0	Modo
0	0	Interrupción por commutación
0	1	Reservado
1	0	Flanco de bajada en ACo
1	1	Flanco de subida en ACo

Registro DIDR1. Registro 1 para deshabilitar la entrada digital.



Bits 7 al 2: No están implementados.

Bits 1 y 0 – AIN1D y AIN2D. Deshabilitan la entrada digital. Las terminales digitales incluyen un *buffer de entrada*, con un 1 en estos bits el buffer queda inhabilitado y por lo tanto, una entrada digital siempre se leería con un 0.

Se sugiere escribir un 1 en estos si se usa el comparador analógico para reducir el consumo de potencia.

El bit **ACME** se encuentra en el Registro ADCSRB (Registro B de Control y Estado del ADC).



Bits ACME (AC Multiplexer Enable). Al ser puesto en alto, la entrada negativa del comparador proviene de la salida del multiplexor del ADC.

El ADC debe estar deshabilitado y el canal de entrada analógica se selecciona con los bits **MUX[3:0]** del registro **ADMUX**.

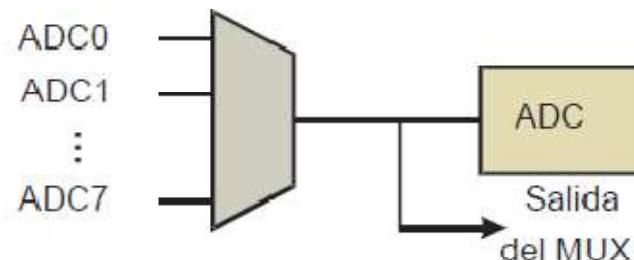
Si el ADC se habilita, aunque **ACME** tenga 1 la entrada será proporcionada en la terminal AIN1.

REG.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	I/O	SRAM
ADCSR _B	-	ACME	-	-	-	ADTS ₂	ADTS ₁	ADTS ₀		(\$7B)

La entrada **AIN₀** se puede reemplazar por un valor constante (V_{BG}). El bit **ACBG** permite hacer esta selección.

	Mínimo	Típico	Máximo
Referencia Interna (V_{BG})	1.0	1.1	1.2

La entrada **AIN₁** se puede reemplazar por una de las entradas para conversión analógico-digital.



Para ello se requiere que:

- El ADC esté deshabilitado (**ADEN** = 0).
- El MUX se habilite para ser usado por el comparador (**ACME** = 1).

Criterios de selección del ADC

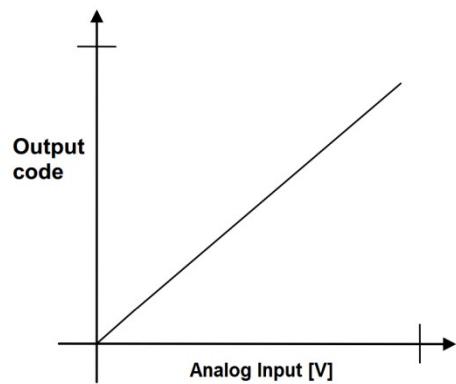
- . Resolución ó nº de bits
- . Tiempo de conversión
- . Nº de canales analógicos de entrada
- . Consumo de potencia

■ Errores en ADC

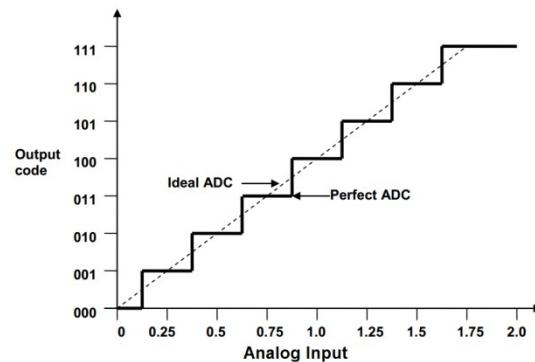
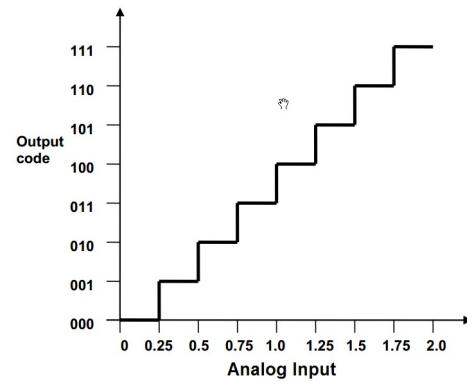
- Los posibles errores de los Adc pueden clasificarse en :
- Cuando la recta de conversión no pasa por el origen → error de offset
- Cuando la pendiente de la recta no es la apropiada → error de ganancia
- Cuando la recta de conversion no es recta → Error de linealidad

ADC

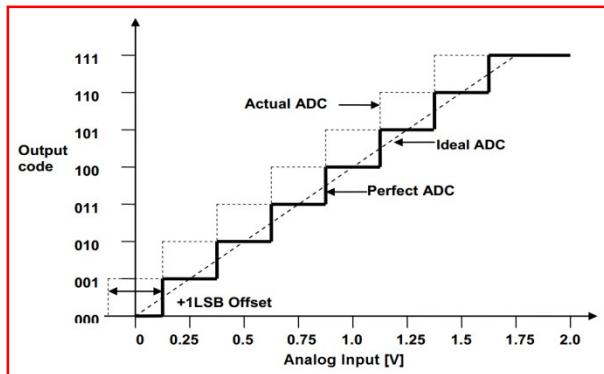
Ideal ADC in Single-ended Mode



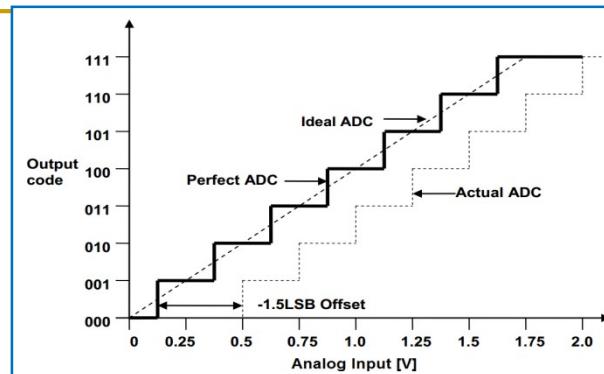
Perfect ADC in Single-ended Mode



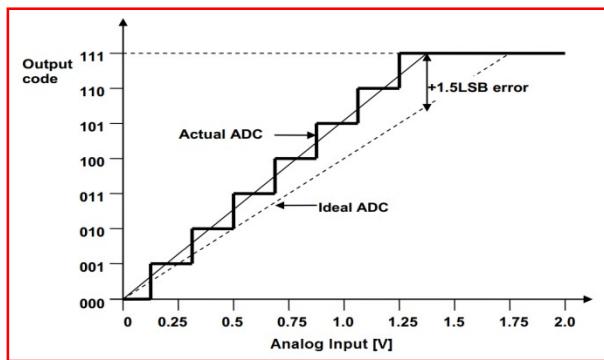
Perfect ADC in Single-ended Mode (Adjusted Quantization)



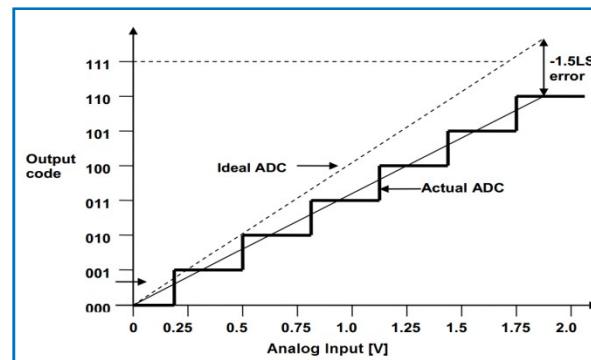
Positive Offset Error



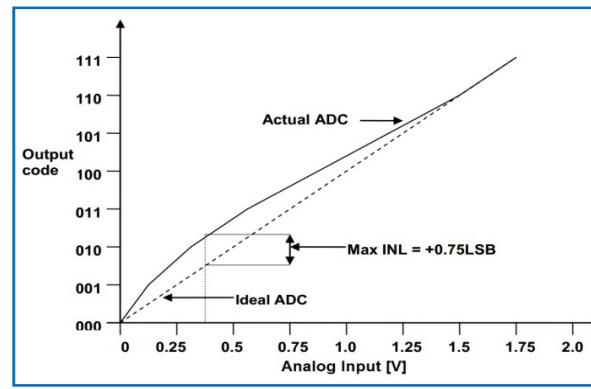
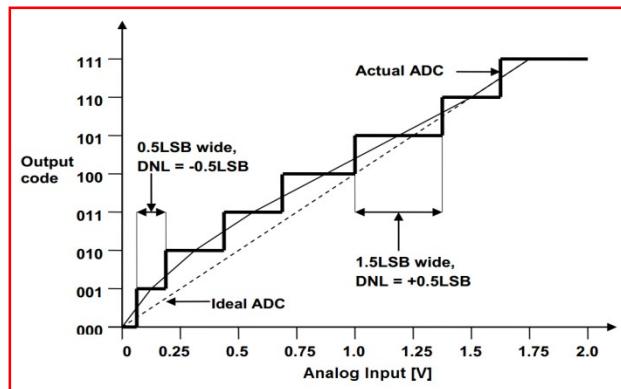
Negative Offset Error



Positive Gain Error



Negative Gain Error

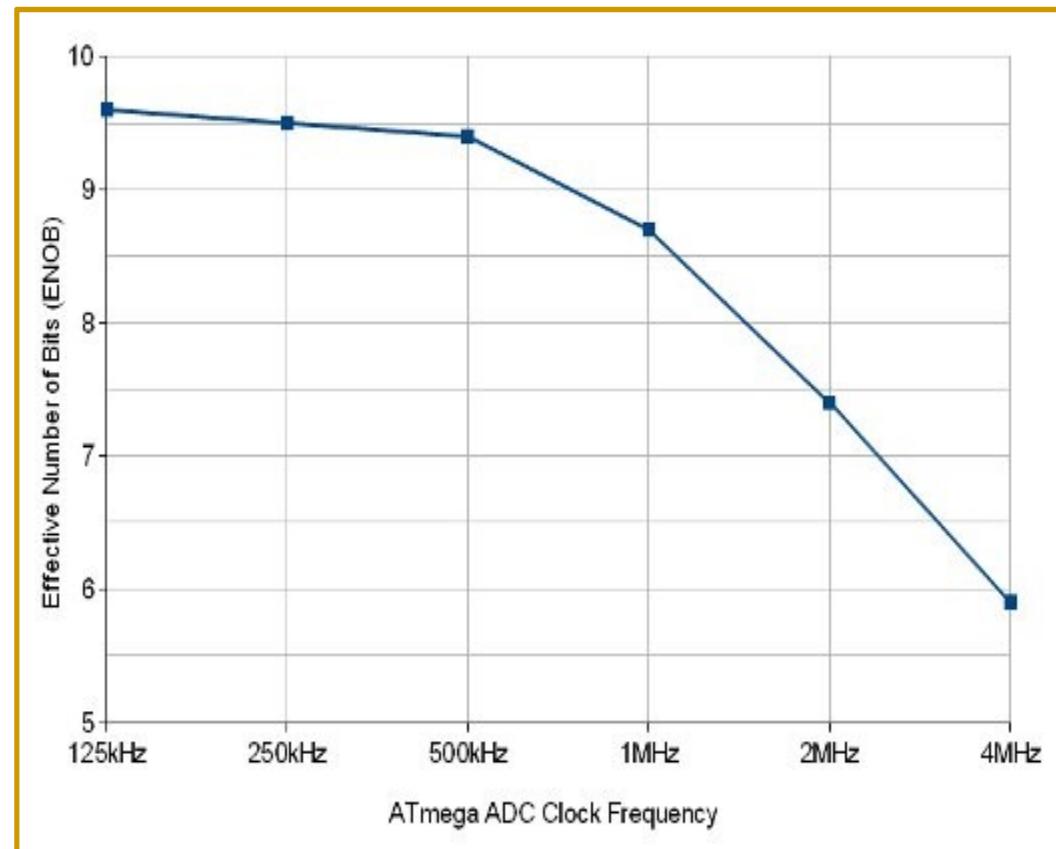


Differential Non- Linearity (DNL)

Integral Non- Linearity (INL)

¿Qué tan rápido *puede* ir el ADC?

- Las hojas de datos de ATmega advierten acerca de no romper el límite de velocidad: ¡reloj ADC de 50kHz a 200kHz para una **resolución máxima!**!
- Entonces, Para probar esto, configuramos un circuito para muestrear un tono sinusoidal puro con su ADC



Relación señal a ruido (SNR)

- SNR se define como la relación entre el nivel de tensión de la señal de salida y el nivel de ruido de salida.
- representado en decibelios (dB) y calculado usando la siguiente fórmula.

$$SNR(dB) = 20 \log \left(\frac{V_{RMS(\text{Signal})}}{V_{RMS(\text{Noise})}} \right)$$

- Por ejemplo, si la amplitud de la señal de salida es 1V (RMS) y la amplitud de ruido de salida es 1mV (RMS), entonces
- el valor de SNR sería 60dB. La fórmula anterior es una definición general de SNR.
- El valor de SNR de un ADC ideal se calcula con la siguiente fórmula:
- $SNR (\text{dB}) = 6.02N + 1.76 (\text{dB})$
- donde N es la resolución (número de bits) del ADC. Por ejemplo, un ADC ideal de 10 bits tendrá una SNR de aproximadamente 62dB

Total Harmonic Distortion (THD)

Cuando una señal de entrada de una frecuencia particular pasa a través de un dispositivo no lineal, contenido adicional se agrega a los armónicos de la frecuencia original.

Por ejemplo, suponga que tiene una señal de entrada de frecuencia f . Entonces las frecuencias armónicas son $2f$, $3f$, $4f$, etc. Así que la no linealidad en el convertidor produce armónicos que no estaban presentes en la señal original.

Estas frecuencias armónicas usualmente distorsionan la salida que degrada el rendimiento del sistema. Este efecto se puede medir usando el término llamado distorsión armónica total (THD).

$$THD = \frac{\sqrt{V_2^2 + V_3^2 + \dots + V_n^2}}{V_1}$$

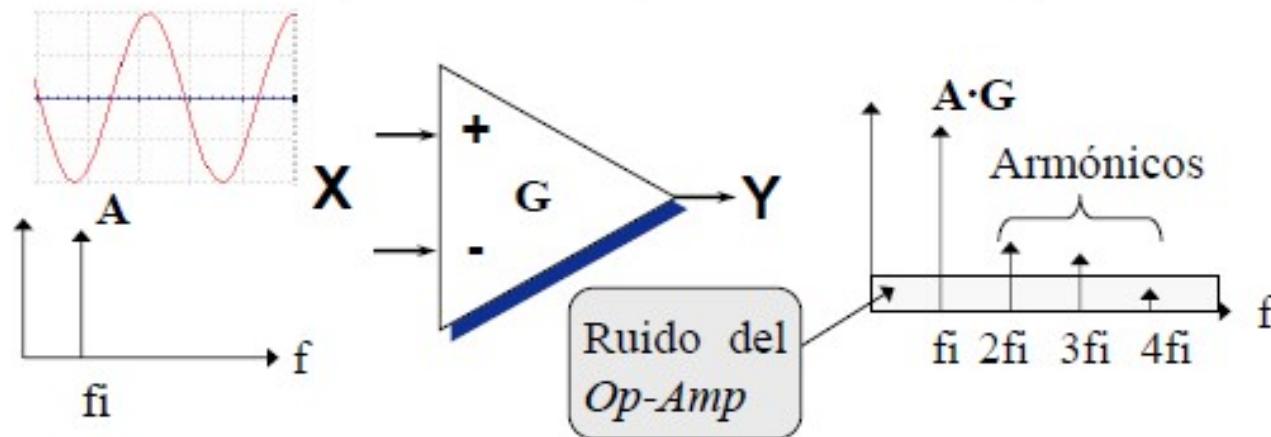
THD se define como la relación entre la suma de potencias de los componentes de frecuencia armónica y la potencia del componente de frecuencia fundamental / original. En términos de voltaje RMS,

El valor THD aumenta con el aumento en la frecuencia. El THD debe tener un valor mínimo para menos distorsión. A medida que la amplitud de la señal de entrada aumenta, la distorsión también aumenta.

Distorsión Armónica Total mas Ruido (THD+N)

Se define como la relación entre el valor *RMS* de la tensión de ruido más el valor *RMS* de la tensión de los armónicos referido al valor *RMS* de la componente fundamental:

$$THD + N = \left[\frac{(\sum RMS \text{ armónicos} + RMS \text{ ruido})}{RMS \text{ fundamental}} \right] \times 100 \%$$



SRN y distorsión (SINAD)

- La señal al ruido y la distorsión (SINAD) es una combinación de parámetros SNR y THD. Se define como el
- relación entre el valor RMS de la amplitud de señal y el valor RMS de todos los demás componentes espectrales,
- incluyendo armónicos, pero excluyendo la componente continua (DC).

$$SINAD = -10 \log(10^{-SNR/10} + 10^{-THD/10})$$

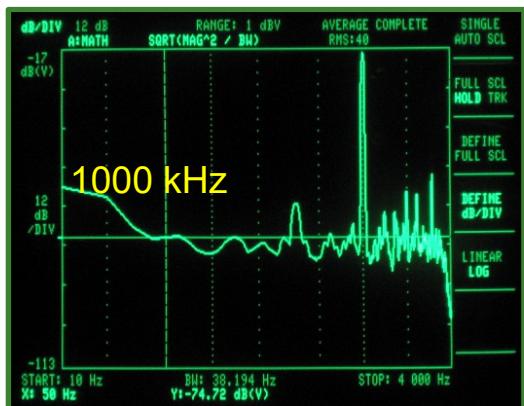
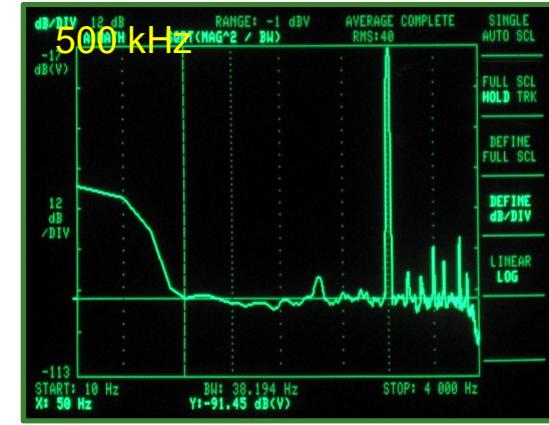
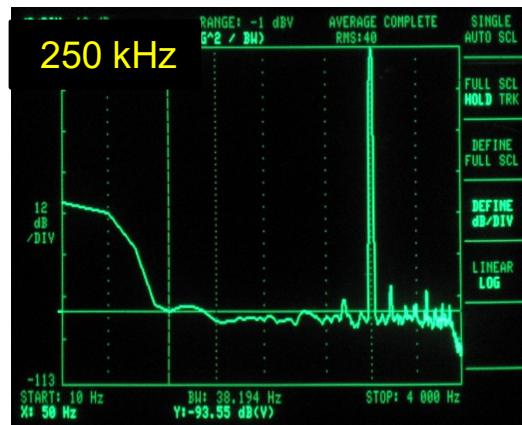
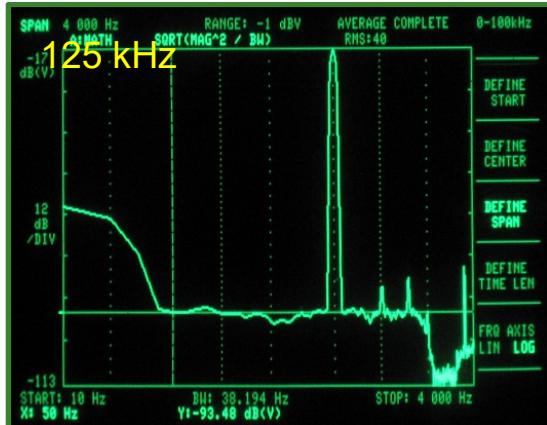
- Para representar el rendimiento dinámico general de un ADC, SINAD es una buena opción ya que incluye componentes de ruido y distorsión.

Número efectivo de bits (ENOB)

- El número efectivo de bits (ENOB) es la cantidad de bits con los que el ADC se comporta como un ADC perfecto.
- Eso es otra forma de representar la relación señal / ruido y la distorsión (SINAD) y puede derivarse de la siguiente fórmula.

$$ENOB = \frac{SINAD - 1.76}{6.02}$$

Análisis del espectro ADC



$$SINAD = -10 \log \left(10^{-SNR/10} + 10^{-THD/10} \right)$$

Para calcular el número efectivo de bits (ENOB), necesitamos saber 3 cosas:

- (1) el nivel de la señal,
- (2) el piso de ruido, y
- (3) el ancho de banda total del piso de ruido.

Esto nos permitirá calcular la relación señal / ruido (SNR), que es, como suena, el nivel de señal dividido por el nivel de ruido total. A partir de ahí, usamos la siguiente fórmula para calcular la ENOB:

$$ENOB = \frac{SINAD - 1.76}{6.02}$$