BME KAUT

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésautomatikai Tanszék

Mechatronics and Microcomputers

Stipendium Hungaricum

2018/2019 Autumn Semester

Szilárd Aradi, PhD

Hardware Interrupts (IT)

- An asynchronous signal to the processor emitted by a hardware peripheral indicating an event that needs immediate attention.
- It alerts the CPU to a condition requiring interruption of the current code the processor is executing. The CPU responds by suspending its current activities, saving its state to the stack, and executing a function called an interrupt handler (or an interrupt service routine, ISR) to deal with the event. This interruption is temporary, and, after the interrupt handler finishes, the CPU resumes normal activities.

Interrupt Sources and Vectors

- The AVR provides several different interrupt sources. (In case of Atmega128 the number of IT sources is 35.)
- All interrupts are assigned individual enable bits which must be written logic one together with the Global Interrupt Enable bit in the Status Register in order to enable the interrupt.
- The lowest addresses in the program memory space are by default defined as the Reset and Interrupt vectors.
- The place determines the priority levels of the different interrupts. The lower the address the higher is the priority level. RESET has the highest priority.
- The interrupt vectors can be moved to the start of the boot Flash section by setting the IVSEL bit in the MCU Control Register (MCUCR). The Reset vector can also be moved to the start of the boot Flash section by programming the BOOTRST fuse. It supports the bootloader development.

Interrupt handling

- When an interrupt occurs, the Global Interrupt Enable I-bit is cleared and all interrupts are disabled. The user software can write logic one to the I-bit to enable nested interrupts. All enabled interrupts can then interrupt the current interrupt routine. The I-bit is automatically set when a Return from Interrupt instruction – RETI – is executed.
- There are basically two types of interrupts.
  - The first type is triggered by an event that sets the interrupt flag. For these interrupts, the Program Counter is vectored to the actual interrupt vector in order to execute the interrupt handling routine, and hardware clears the corresponding interrupt flag.
  - The second type of interrupts will trigger as long as the interrupt condition is present. These interrupts do not necessarily have interrupt flags. If the interrupt condition disappears before the interrupt is enabled, the interrupt will not be triggered.
- When the AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.
- Before critical program code (e.g. EEPROM writing) it is necessary to disable the interrupts with the CLI instruction. In this case the interrupts will be immediately disabled. No interrupt will be executed after the CLI instruction, even if it occurs simultaneously with the CLI instruction.

Interrupt Response Time

- The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum.
- During this 4-clock cycle period, the Program Counter is pushed onto the Stack. The vector is normally a jump to the interrupt routine, and this jump takes three clock cycles.
- If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served.
- A return from an interrupt handling routine takes four clock cycles. During these 4-clock cycles, the Program Counter (two bytes) is popped back from the Stack, the Stack Pointer is incremented by two, and the I-bit in SREG is set.

Interrupt Vectors I.

No.	Addr.	Source	Description
1	$0000	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	$0002	INT0	External Interrupt Request 0
3	$0004	INT1	External Interrupt Request 1
4	$0006	INT2	External Interrupt Request 2
5	$0008	INT3	External Interrupt Request 3
6	$000A	INT4	External Interrupt Request 4
7	$000C	INT5	External Interrupt Request 5
8	$000E	INT6	External Interrupt Request 6
9	$0010	INT7	External Interrupt Request 7
10	$0012	TIMER2 COMP	Timer/Counter2 Compare Match

Interrupt Vectors II.

No.	Addr.	Source	Description
11	$0014	TIMER2 OVF	Timer/Counter2 Overflow
12	$0016	TIMER1 CAPT	Timer/Counter1 Capture Event
13	$0018	TIMER1 COMPA	Timer/Counter1 Compare Match A
14	$001A	TIMER1 COMPB	Timer/Counter1 Compare Match B
15	$001C	TIMER1 OVF	Timer/Counter1 Overflow
16	$001E	TIMER0 COMP	Timer/Counter0 Compare Match
17	$0020	TIMER0 OVF	Timer/Counter0 Overflow
18	$0022	SPI, STC	SPI Serial Transfer Complete
19	$0024	USART0, RX	USART0, Rx Complete
20	$0026	USART0, UDRE	USART0 Data Register Empty

Interrupt Vectors III.

Ssz.	Cím	Forrás	Leírás
21	$0028	USART0, TX	USART0, Tx Complete
22	$002A	ADC	ADC Conversion Complete
23	$002C	EE READY	EEPROM Ready
24	$002E	ANALOG COMP	Analog Comparator event
25	$0030	TIMER1 COMPC	Timer/Countre1 Compare Match C
26	$0032	TIMER3 CAPT	Timer/Counter3 Capture Event
27	$0034	TIMER3 COMPA	Timer/Counter3 Compare Match A
28	$0036	TIMER3 COMPB	Timer/Counter3 Compare Match B
29	$0038	TIMER3 COMPC	Timer/Counter3 Compare Match C
30	$003A	TIMER3 OVF	Timer/Counter3 Overflow

Interrupt Vectors IV.

Ssz.	Cím	Forrás	Leírás
31	$003C	USART1, RX	USART1, Rx Complete
32	$003E	USART1, UDRE	USART1 Data Register Empty
33	$0040	USART1, TX	USART1, Tx Complete
34	$0042	TWI	Two-wire Serial Interface
35	$0044	SPM READY	Store Program Memory Ready

Typical program setup

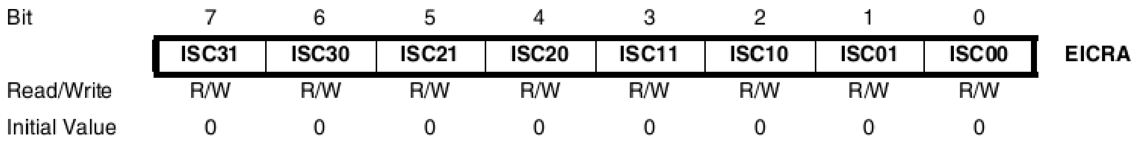
```
$0000 jmp INIT                  ; Reset handler
$0002 jmp EXT_INT0              ; External Interrupt 0 handler
…                               ; Further interrupt handler routines

$0046 INIT: ldi R16, high(RAMEND) ; Starting with the stack configuration
            out SPH, R16           ; Setting the stack pointer low-byte
            ldi R16, low(RAMEND) ; and high-byte
            out SPL, R16
            sei                 ; Global Interrupt Enable
```

External Interrupts I.

- The External Interrupts are triggered by the INT7:0 pins.
  - E.g. push-button handling
- The External Interrupts can be triggered by a falling or rising edge or a low level.
  - When the external interrupt is enabled and is configured as level triggered, the interrupt will trigger as long as the pin is held low.
- Low level interrupts and the edge interrupt on INT3:0 are detected asynchronously. This implies that these interrupts can be used for waking the part also from sleep modes other than Idle mode.
  - If a level triggered interrupt is used for wake-up from Power-down mode, the changed level must be held for some time to wake up the MCU. This makes the MCU less sensitive to noise.

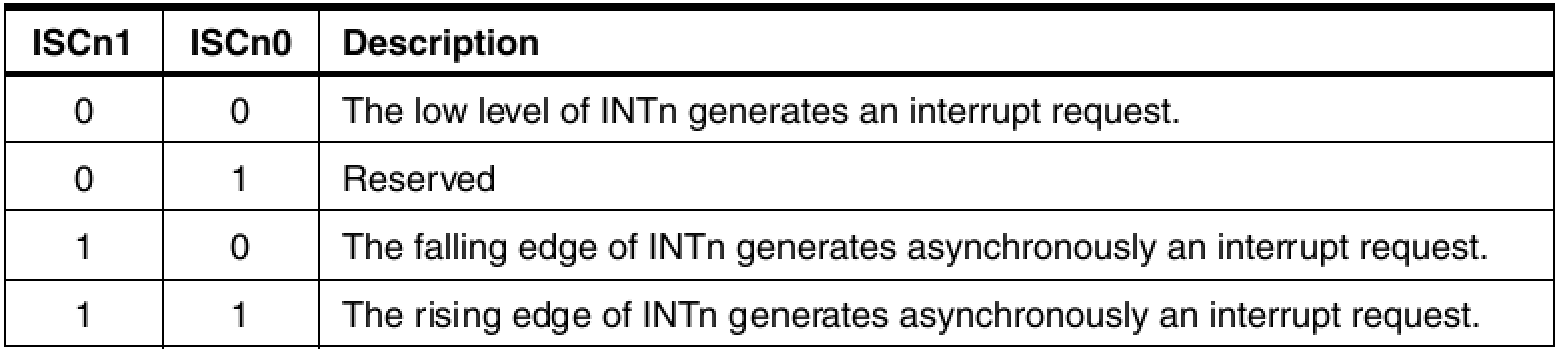
External Interrupts II.

- External Interrupt Control Register A (EICRA)



Bit	7	6	5	4	3	2	1	0	
	ISC31	ISC30	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00	EICRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

  - External Interrupt 3 - 0 Sense Control Bits

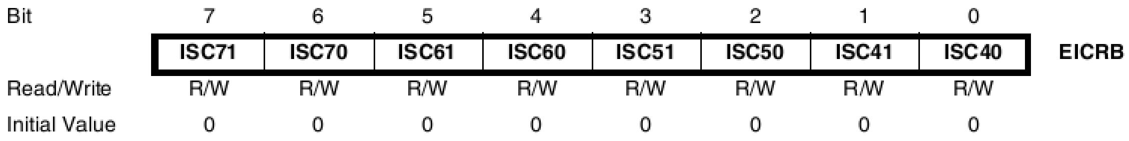


ISCn1	ISCn0	Description
0	0	The low level of INTn generates an interrupt request.
0	1	Reserved
1	0	The falling edge of INTn generates asynchronously an interrupt request.
1	1	The rising edge of INTn generates asynchronously an interrupt request.

  - Minimum pulse width in case of edge-triggering: 50 ns

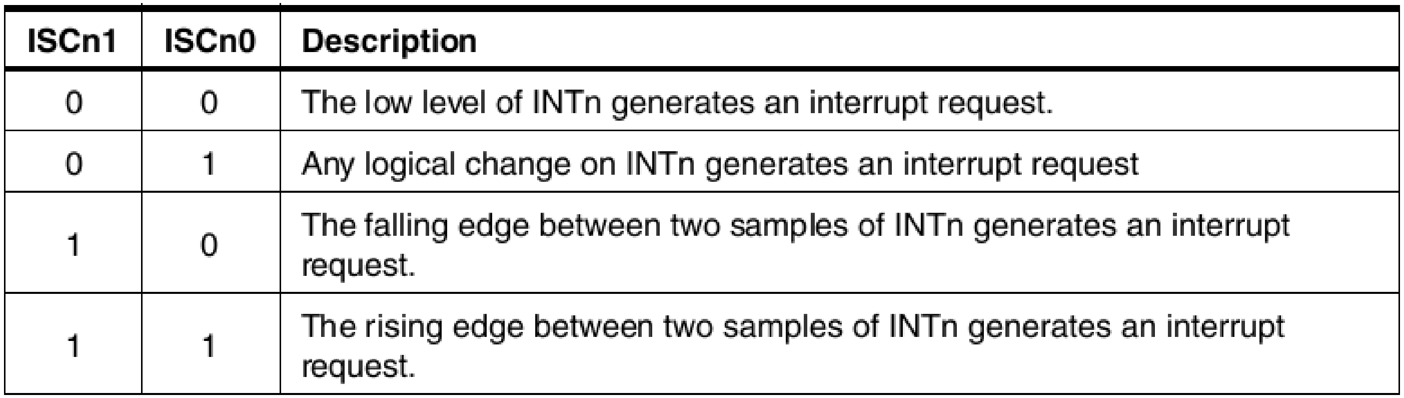
External Interrupts III.

- External Interrupt Control Register B (EICRB)



Bit	7	6	5	4	3	2	1	0	
	ISC71	ISC70	ISC61	ISC60	ISC51	ISC50	ISC41	ISC40	EICRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

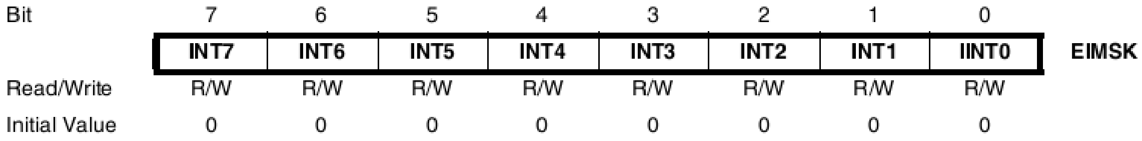
  - External Interrupt 7 - 4 Sense Control Bits



ISCn1	ISCn0	Description
0	0	The low level of INTn generates an interrupt request.
0	1	Any logical change on INTn generates an interrupt request
1	0	The falling edge between two samples of INTn generates an interrupt request.
1	1	The rising edge between two samples of INTn generates an interrupt request.

External Interrupts IV.

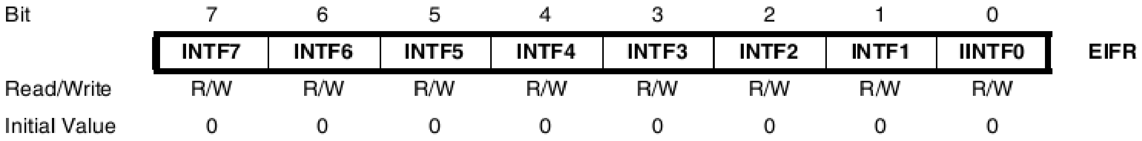
- External Interrupt Mask Register (EIMSK)



Bit	7	6	5	4	3	2	1	0	
	INT7	INT6	INT5	INT4	INT3	INT2	INT1	IINT0	EIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

  - External Interrupt Request 7-0 Enable bits

- External Interrupt Flag Register (EIFR)



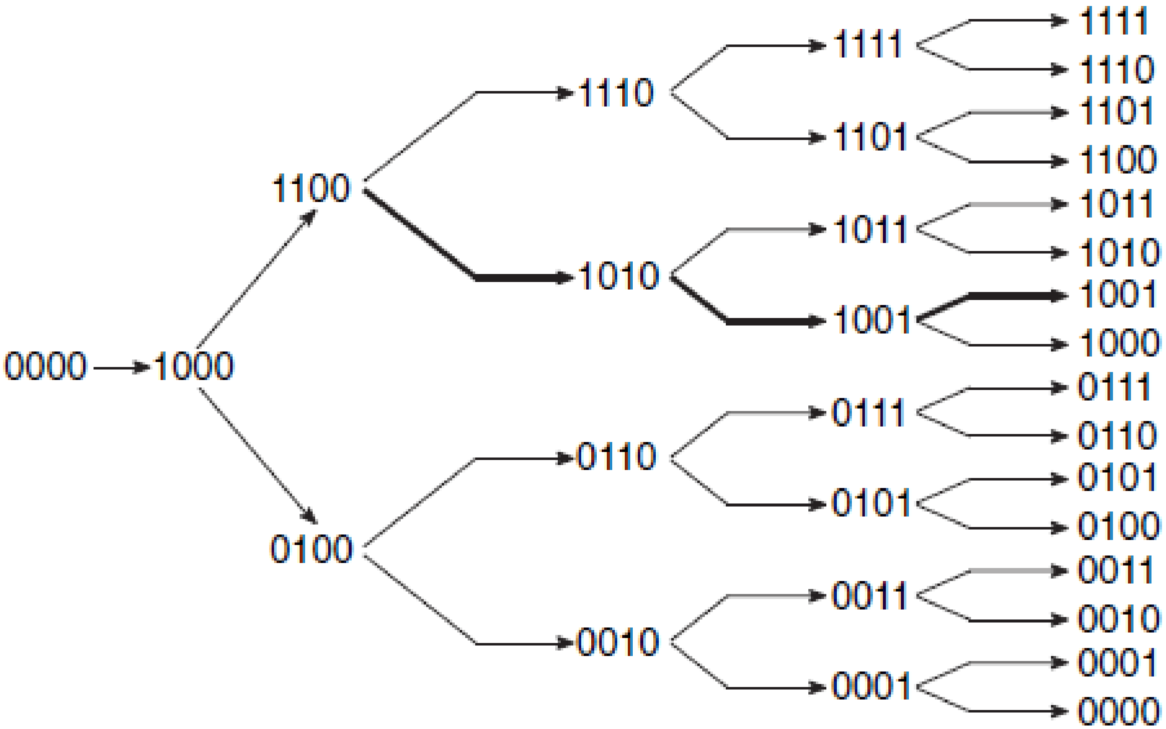
Bit	7	6	5	4	3	2	1	0	
	INTF7	INTF6	INTF5	INTF4	INTF3	INTF2	INTF1	IINTF0	EIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

  - External Interrupt Flags 7-0

A/D Converter I.

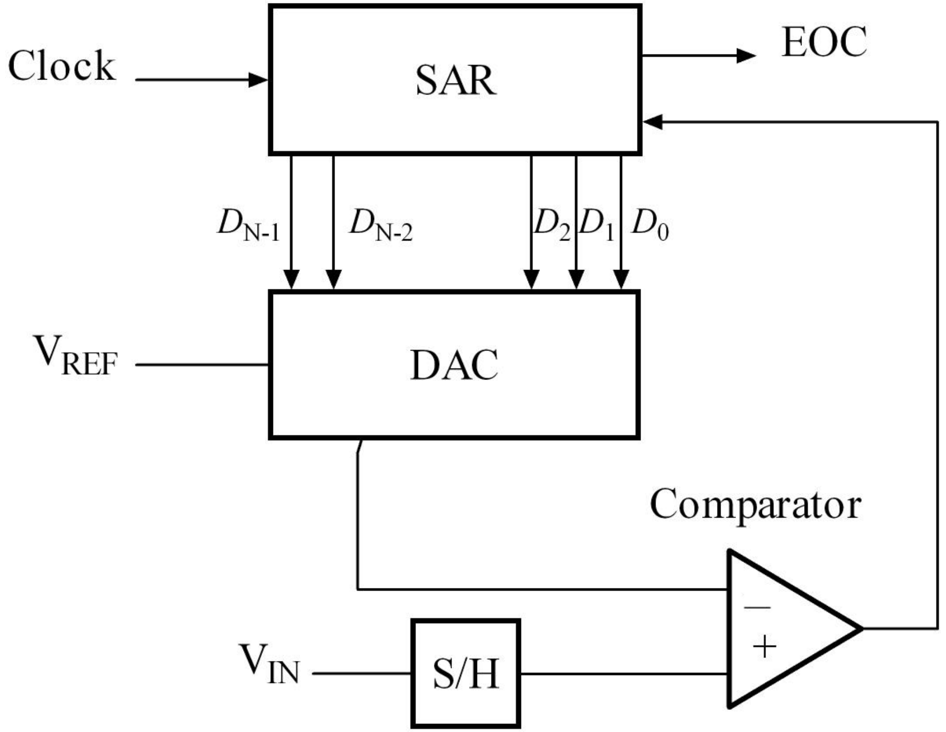
- Features and properties
  - **10-bit resolution**
  - 13-260 µs Conversion Time
  - **Up to. 76.9 kS/s (Up to 15 kS/s at max. resolution) sampling frequency**
  - **8 Multiplexed Single Ended Input Channels**
  - 7 Differential Input Channels (with common grund)
  - 2 Differential Input Channels with Optional Gain of 10x and 200x
  - 0 - VCC ADC Input Voltage Range
  - Selectable 2.56V ADC Reference Voltage
  - Free Running or Single Conversion Mode
  - Interrupt on ADC Conversion Complete

Successive Approximation



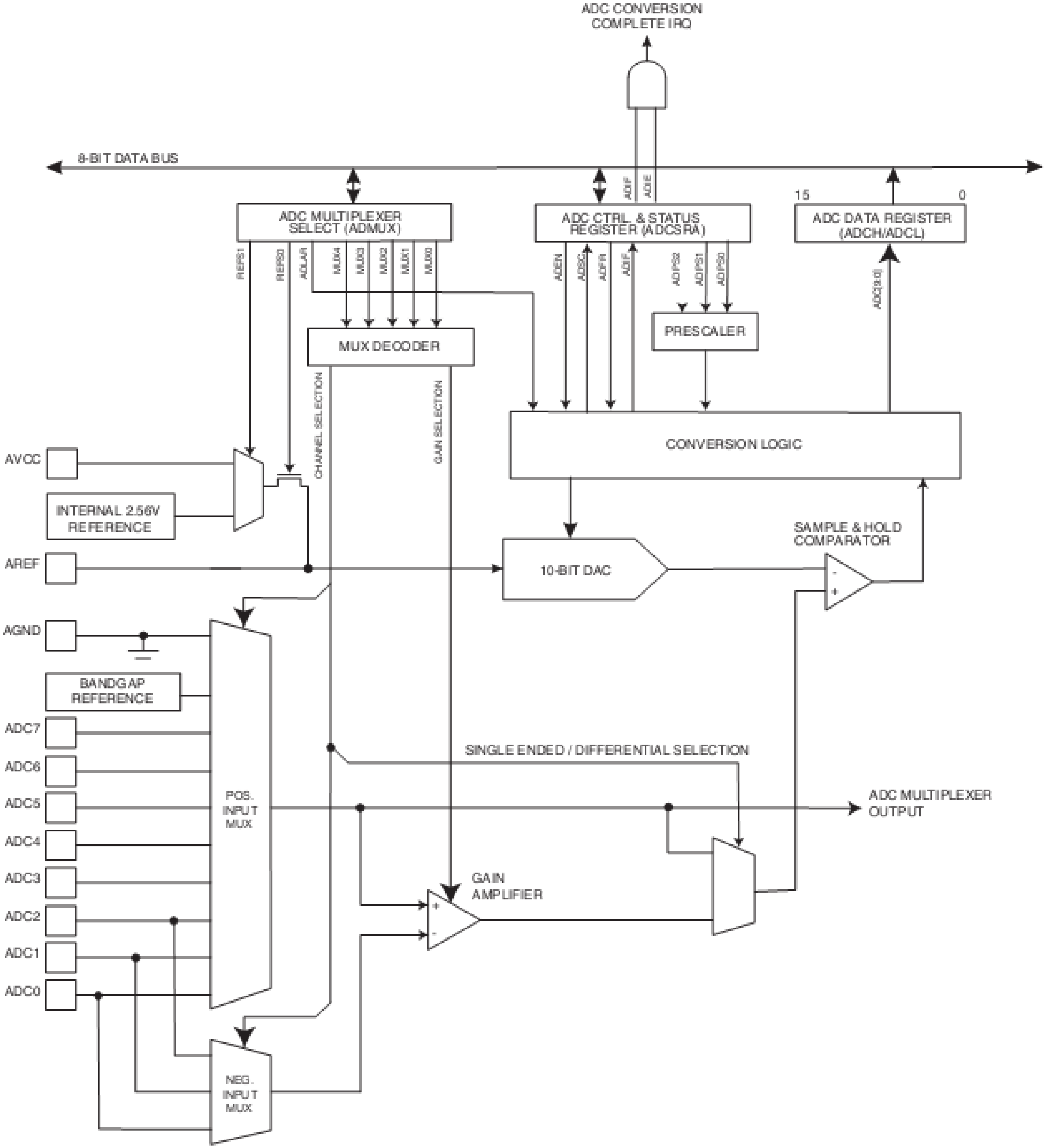
Conversion process in a successive approximation type A/D converter.

A/D Converter Operation



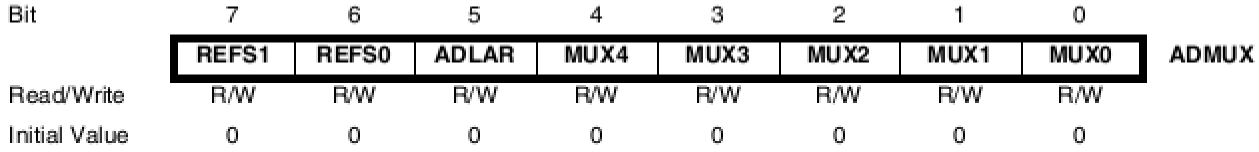
By White Flye - Own work, CC BY-SA 2.5, https://commons.wikimedia.org/w/index.php?curid=37953205

- DAC = digital-to-analog converter
- EOC = end of conversion
- SAR = successive approximation register
- S/H = sample and hold circuit
- Vin = input voltage
- Vref = reference voltage


ADC CONVERSION COMPLETE IRQ
8-BIT DATA BUS
ADIF
ADIE
15
0
ADC MULTIPLEXER SELECT (ADMUX)
ADC CTRL. & STATUS REGISTER (ADCSRA)
ADC DATA REGISTER (ADCH/ADCL)
REFS1
REFS0
ADLAR
MUX4
MUX3
MUX2
MUX1
MUX0
ADEN
ADSC
ADFR
ADIF
ADPS2
ADPS1
ADPS0
ADC[9:0]
MUX DECODER
PRESCALER
CHANNEL SELECTION
GAIN SELECTION
CONVERSION LOGIC
AVCC
INTERNAL 2.56V REFERENCE
AREF
10-BIT DAC
SAMPLE & HOLD COMPARATOR
AGND
BANDGAP REFERENCE
ADC7
ADC6
ADC5
ADC4
ADC3
ADC2
ADC1
ADC0
POS. INPUT MUX
NEG. INPUT MUX
SINGLE ENDED / DIFFERENTIAL SELECTION
ADC MULTIPLEXER OUTPUT
GAIN AMPLIFIER

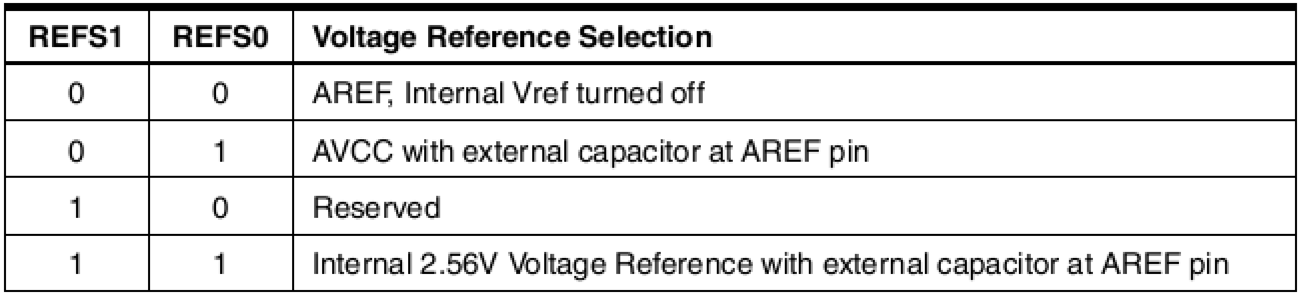
A/D Registers I.

- ADC Multiplexer Selection Register (ADMUX)



Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

  - Reference Selection Bits (REFS 1:0)

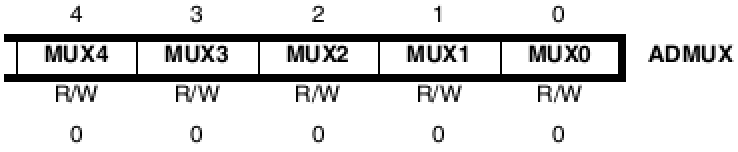


REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal Vref turned off
0	1	AVCC with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 2.56V Voltage Reference with external capacitor at AREF pin

  - ADC Left Adjust Result (ADLAR)

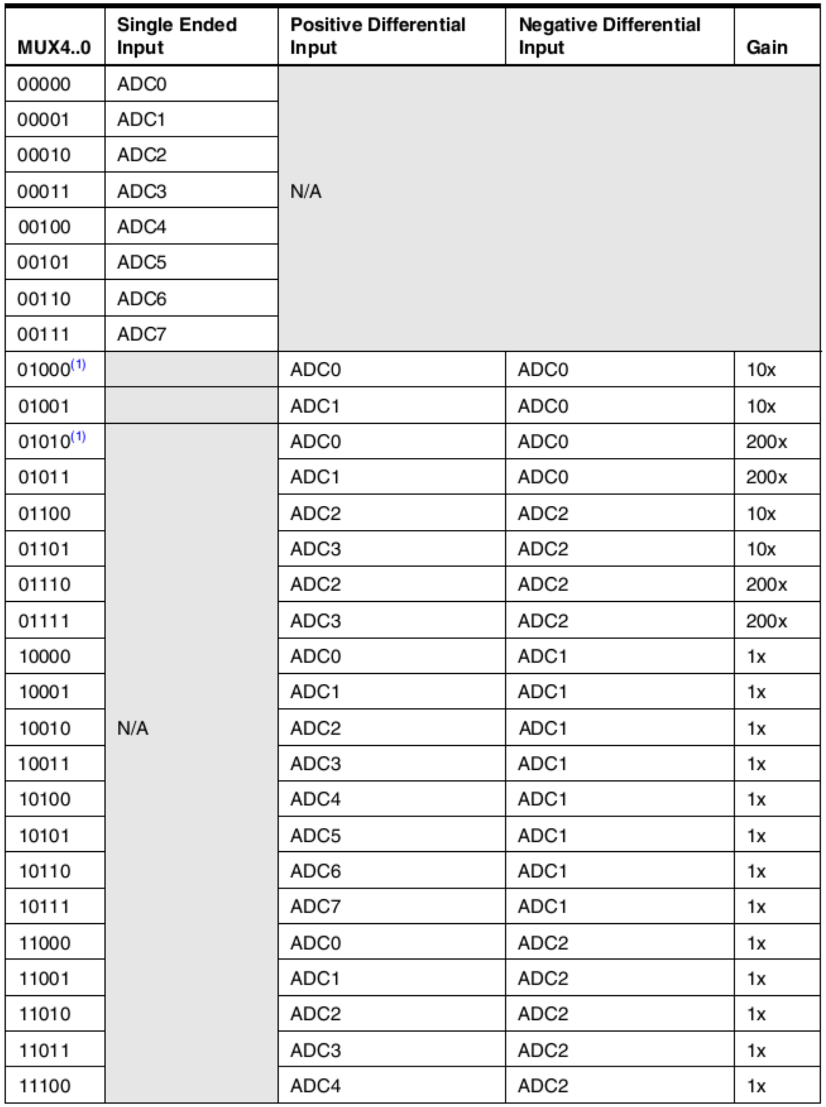
A/D Registers II.

- ADMUX

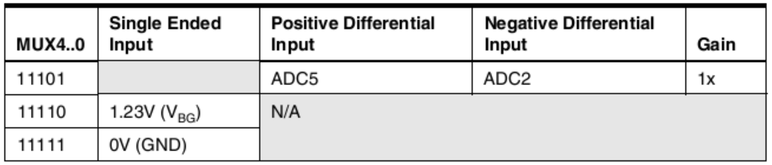


	4	3	2	1	0	
	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
	R/W	R/W	R/W	R/W	R/W	
	0	0	0	0	0	

  - Analog Channel and Gain Selection Bits (MUX 4:0)



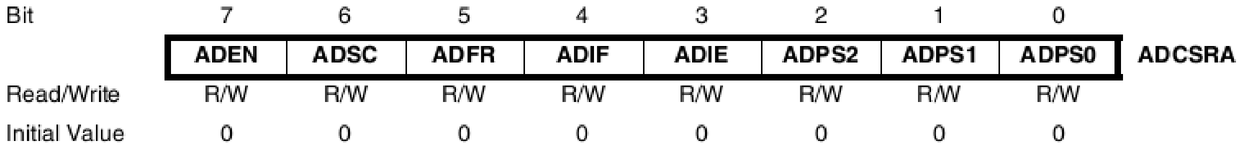
MUX4..0	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain
00000	ADC0	N/A		
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			
01000[1]		ADC0	ADC0	10x
01001		ADC1	ADC0	10x
01010[1]	N/A	ADC0	ADC0	200x
01011		ADC1	ADC0	200x
01100		ADC2	ADC2	10x
01101		ADC3	ADC2	10x
01110		ADC2	ADC2	200x
01111		ADC3	ADC2	200x
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
11000		ADC0	ADC2	1x
11001		ADC1	ADC2	1x
11010		ADC2	ADC2	1x
11011		ADC3	ADC2	1x
11100		ADC4	ADC2	1x



MUX4..0	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain
11101		ADC5	ADC2	1x
11110	1.23V (V_{BG})	N/A		
11111	0V (GND)			

A/D Registers III.

- ADC Control and Status Register A (ADCSRA)

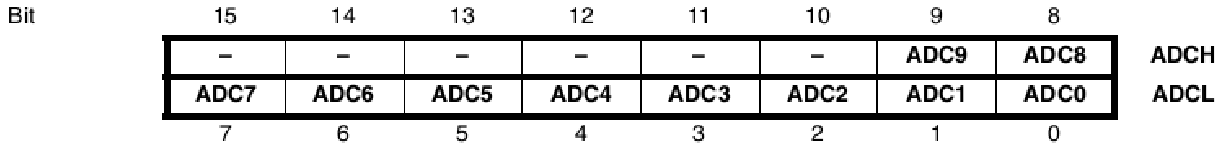


Bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

  - ADC Enable (ADEN)
  - ADC Start Conversion (ADSC)
  - ADC Free Running Select (ADFR)
  - ADC Interrupt Flag (ADIF)
  - ADC Interrupt Enable (ADIE)
  - ADC Prescaler Bits (ADPS 2:0)
    - 1-128, 2^{ADPS}

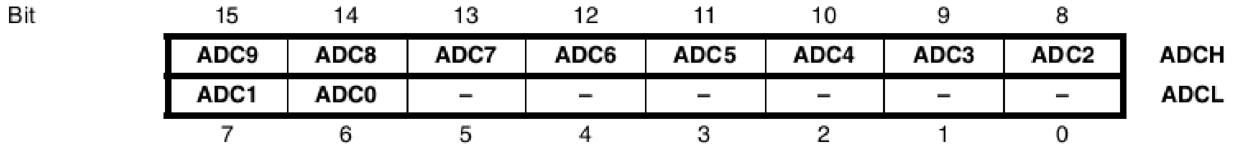
A/D Registers IV.

- ADC Data Register (ADCL, ADCH)
  - ADLAR = 0, right adjusted



Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	–	–	ADC9	ADC8	ADCH
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	

  - ADLAR = 1, left adjusted



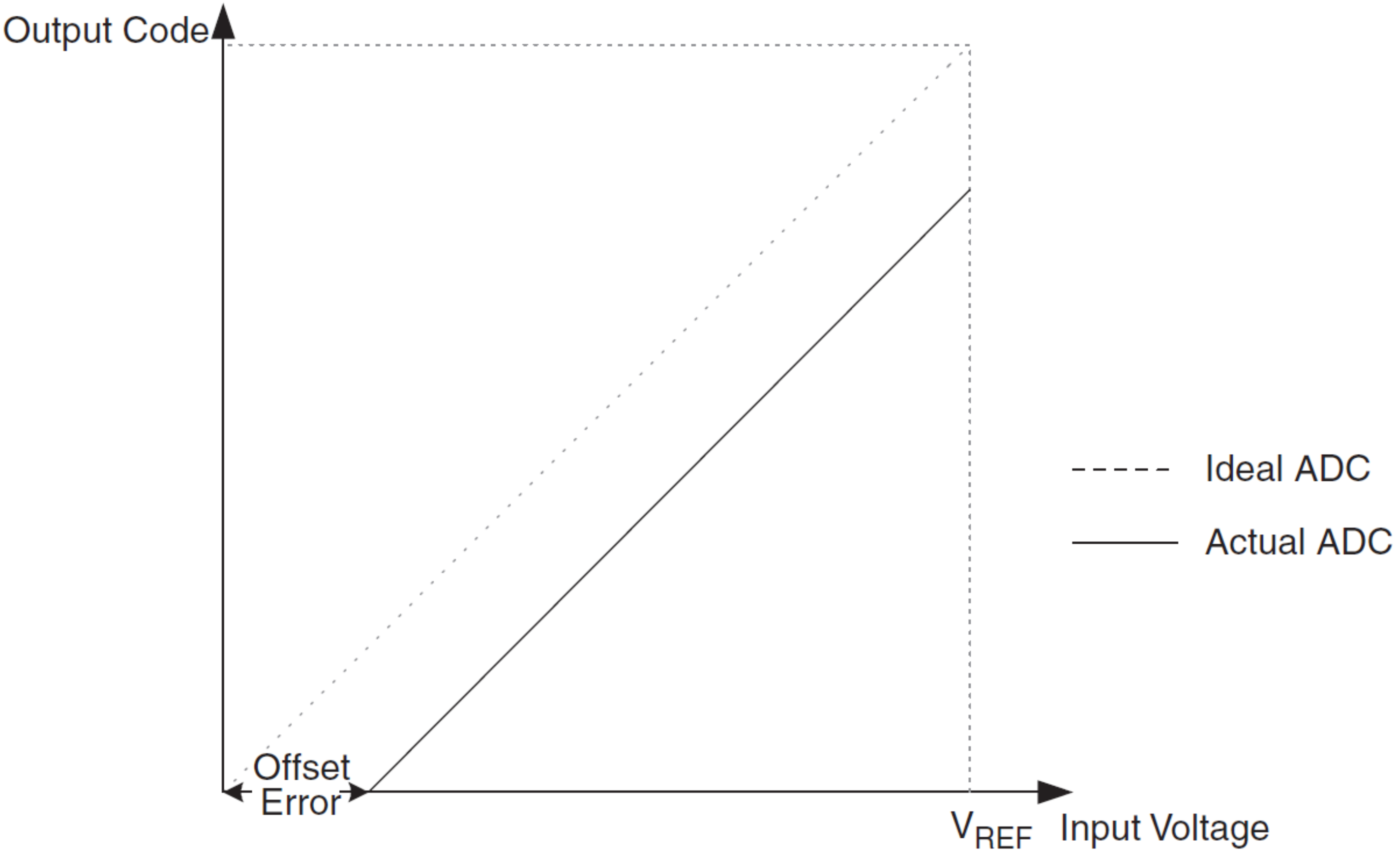
Bit	15	14	13	12	11	10	9	8	
	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	

  - If differential channels are used, the result is presented in two’s complement form.
  - ADCL must be read first, then ADCH.

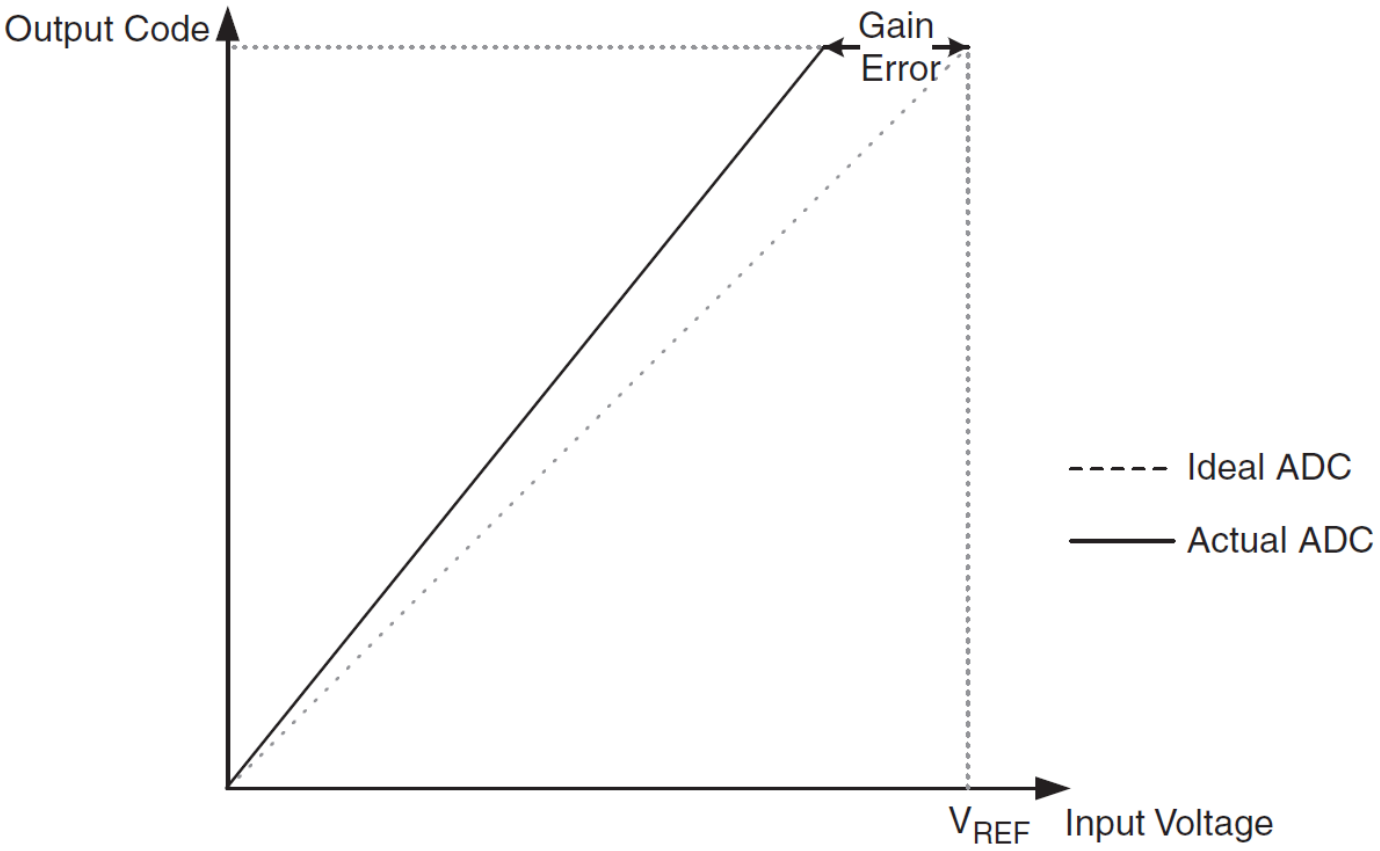
A/D Converters Accuracy

- An n-bit single-ended ADC converts a voltage linearly between GND and V_{REF} in 2^n steps (LSBs). The lowest code is read as 0, and the highest code is read as 2^{n-1}.
- **Offset**: The deviation of the first transition (0x000 to 0x001) compared to the ideal transition.
- **Gain Error**: After adjusting for offset, the gain error is found as the deviation of the last transition (0x3FE to 0x3FF) compared to the ideal transition.
- **Integral Non-linearity** (INL): After adjusting for offset and gain error, the INL is the maximum deviation of an actual transition compared to an ideal transition for any code.
- **Differential Non-linearity** (DNL): The maximum deviation of the actual code width (the interval between two adjacent transitions) from the ideal code width (1LSB).
- **Quantization Error**: Due to the quantization of the input voltage into a finite number of codes, a range of input voltages (1LSB wide) will code to the same value. Always ±0.5LSB.
- **Absolute Accuracy**: The maximum deviation of an actual (unadjusted) transition compared to an ideal transition for any code. This is the compound effect of offset, gain error, differential error, non-linearity, and quantization error.

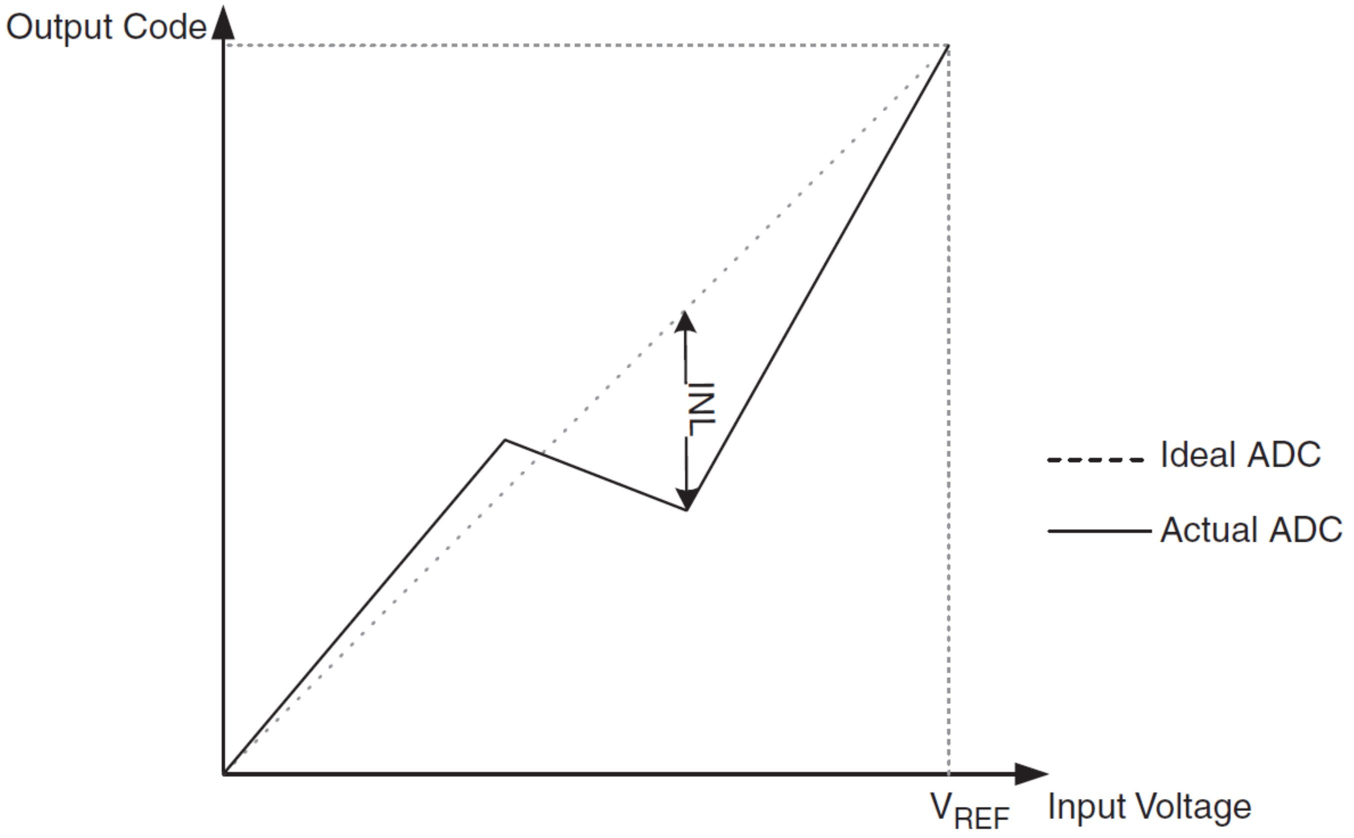
Offset Error



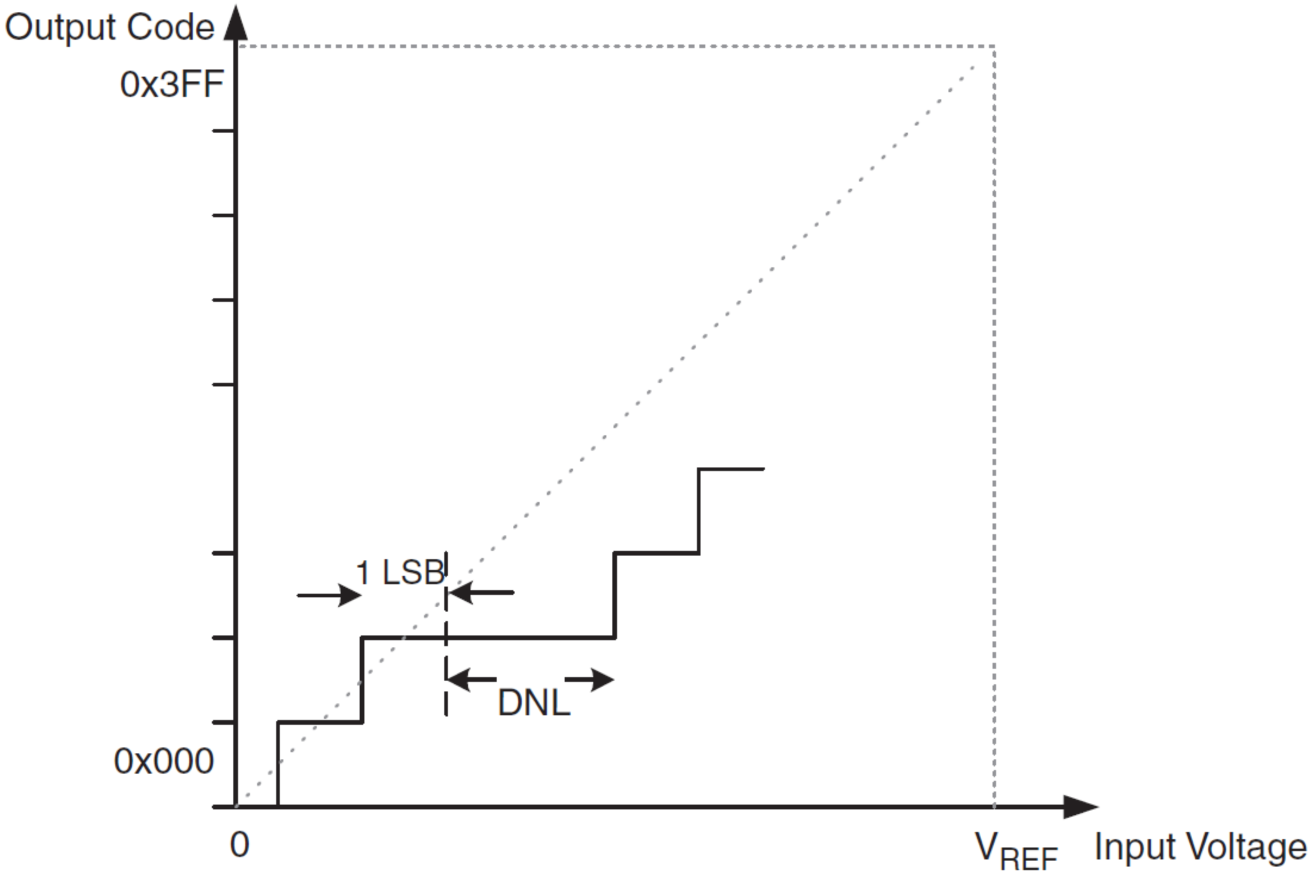
Gain Error


Output Code
Gain Error
Ideal ADC
Actual ADC
V_{REF}
Input Voltage

INL



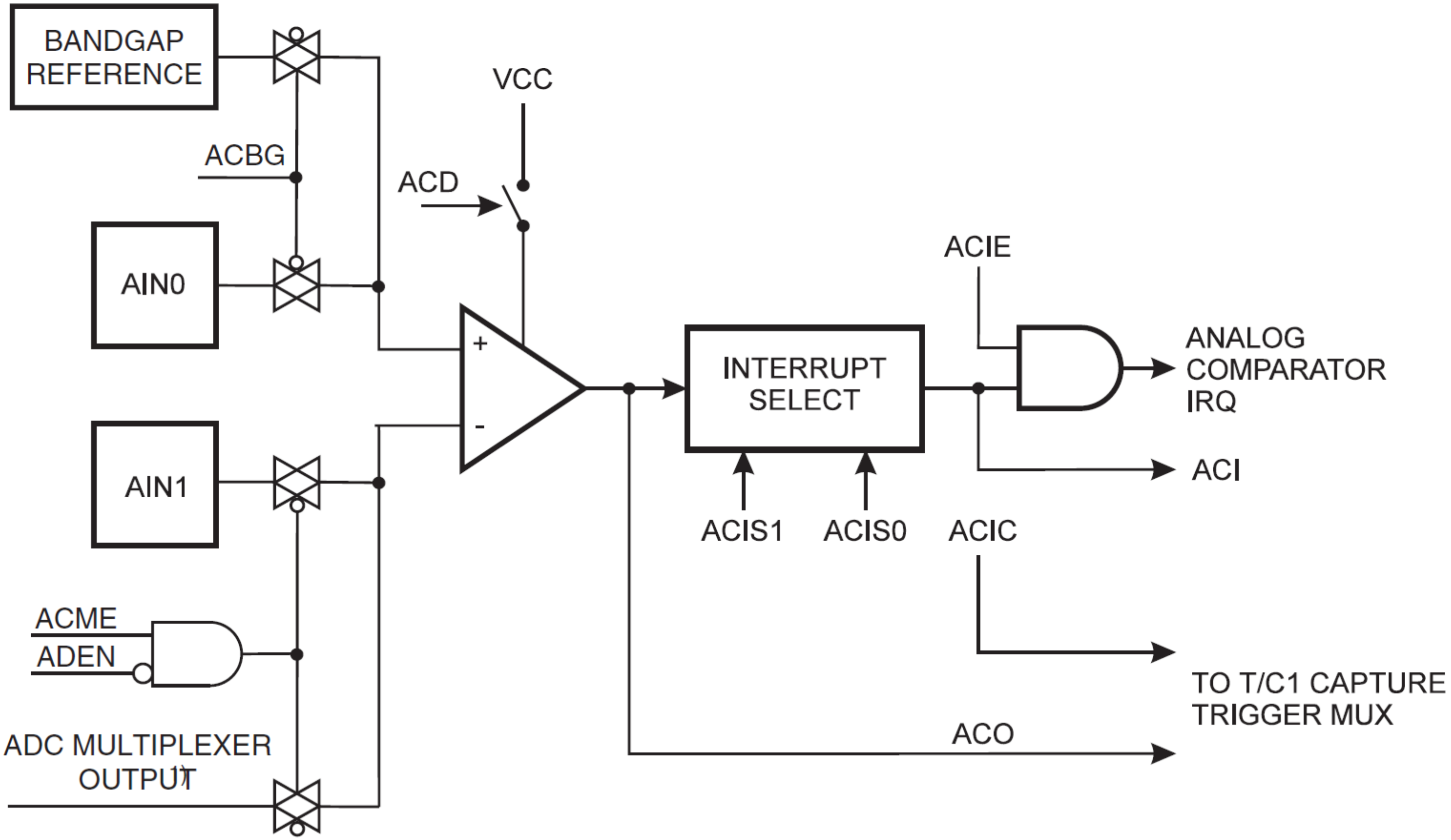
DNL



Analog Comparator Operation

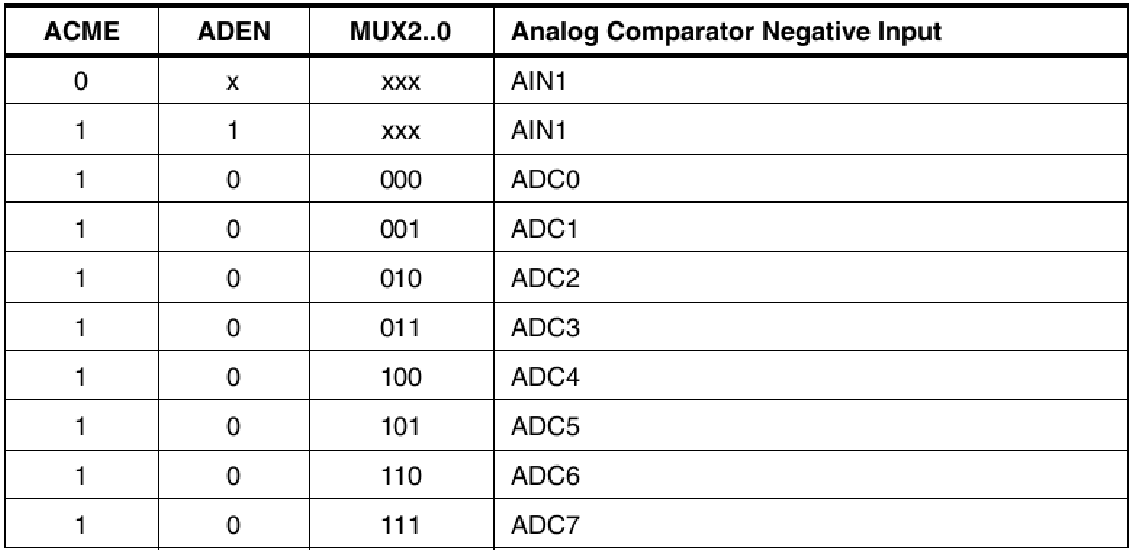
- The Analog Comparator compares the input values on the positive pin AIN0 and negative pin AIN1.
  - When the voltage on the positive pin AIN0 is higher than the voltage on the negative pin AIN1, the Analog Comparator Output, ACO, is set.
  - The comparator's output can be set to trigger the Timer/Counter1 Input Capture function.

Analog Comparator Architecture



Analog Comparator Registers

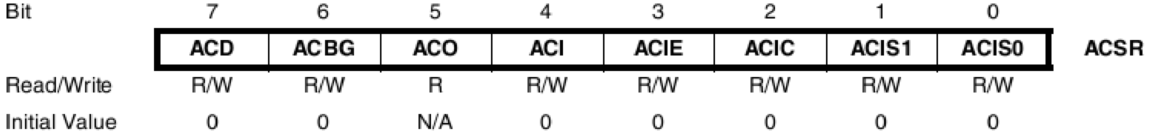
- Analog Comparator Multiplexer Enable (ACME) bit (SFIOR bit 3)
  - When this bit is written logic zero, AIN1 is applied to the negative input of the Analog Comparator.
  - When this bit is written logic one and the ADC is switched off, the ADC multiplexer selects the negative input to the Analog Comparator.



ACME	ADEN	MUX2..0	Analog Comparator Negative Input
0	x	xxx	AIN1
1	1	xxx	AIN1
1	0	000	ADC0
1	0	001	ADC1
1	0	010	ADC2
1	0	011	ADC3
1	0	100	ADC4
1	0	101	ADC5
1	0	110	ADC6
1	0	111	ADC7

Analog Comparator Registers

- Analog Comparator Control and Status Register (ACSR)



Bit	7	6	5	4	3	2	1	0	
	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	N/A	0	0	0	0	0	

  - Analog Comparator Disable (ACD)
  - Analog Comparator Bandgap Select (ACBG): V_{BG}=1,23V
  - Analog Comparator Output (ACO)
  - Analog Comparator Interrupt Flag (ACI)
  - Analog Comparator Interrupt Enable (ACIE)
  - Analog Comparator Input Capture Enable (ACIC)
  - Analog Comparator Interrupt Mode Select (ACIS 1:0)
    - 00: IT on Output Toggle, 01: Reserved, 10: IT on Falling Output Edge, 11: IT on Rising Output Edge

The End

Thank you for your attention!