Mechatronics Engineering

Lab 5

Analog-to-Digital Converters (ADC)

GUC German University in Cairo

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Tutorial Contents

- Analog-to-Digital Conversion
- Analog-to-Digital Converter (ADC) in ATmega328P
  - Features
  - Associated Registers and Timing Diagram
- Lab 5 Validation

Tutorial Contents

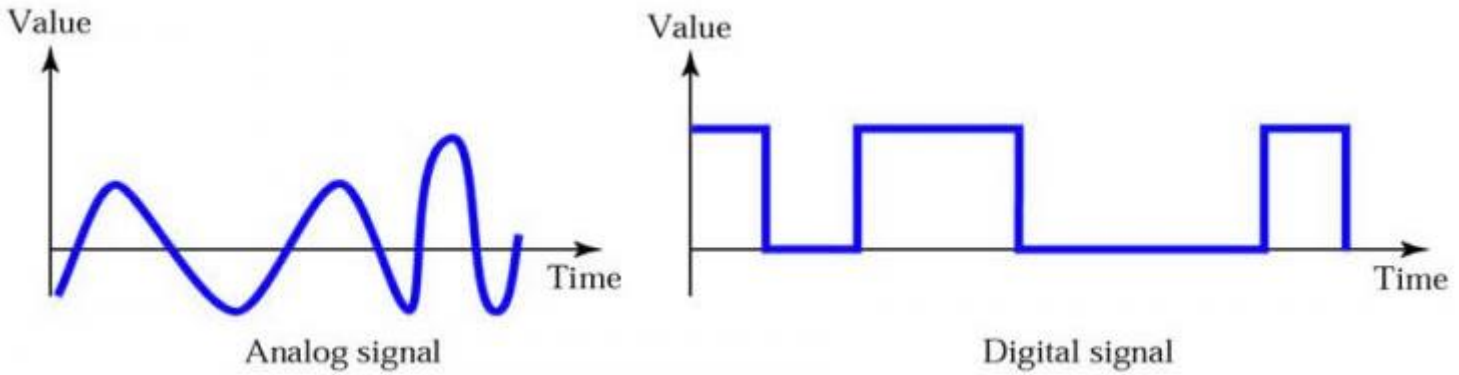
- **Analog-to-Digital Conversion**

- Analog-to-Digital Converter (ADC) in ATmega328P
  - Features
  - Associated Registers and Timing Diagram

- Lab 5 Validation

Analog-to-Digital Conversion

- Digital computers use binary (discrete) values; however, everything is analog (continuous) in the real physical world.
- Example of analog signals include temperature, pressure and velocity.
- These signals can be obtained using sensors and the analog sensor output needs to be read and processed by the microcontroller.
- Analog-to-Digital Converters (ADC) are used for this purpose – to convert an analog signal to a digital signal which can be processed by the microcontroller.



Analog-to-Digital Conversion

- **Some Characteristics of Analog-to-Digital Converters (ADC)**:
  - Analog Input Channels:
    ADC have multiple analog input channels which are multiplexed. However, the ADC can only convert one analog input channel at a time. For example, an ADC with 6 input channels can have 6 analog sensors connected to it; however, it can convert one sensor at a time and we can select which analog signal to convert through the multiplexer address bits.

  - Digital Data Output:
    An n-bit ADC converts the analog input signal into a digital word of n bits. So, for example, an 8-bit ADC can convert the analog input signal into a digital word of 8 bits width ($0 \rightarrow 255$).

  - Reference Voltage (V_{ref}):
    It is an input voltage used to reference the maximum voltage of the analog input signal. For example, if the analog signal connected to the ADC ranges from 0 to 8 volts, then V_{ref} should be 8 volts.

Analog-to-Digital Conversion

- **Some Characteristics of Analog-to-Digital Converters (ADC)**:
  - Resolution:
    It is a measure of the accuracy of the ADC conversion. It is the smallest step change that can be discerned by the ADC. It is affected by the digital output size (n-bits) and the reference voltage V_{ref}.

$$Resolution = \frac{V_{ref}}{2^n}$$

    For example, an 8-bit ADC with $V_{ref} = 5$ volts has a resolution of:

$$Resolution = \frac{5}{2^8} = 0.01953\ Volts = 19.53\ mV$$

  - Conversion Time:
    It is the time taken by the ADC to convert an analog signal to a digital word. This is affected by the technology of the ADC chip as well as the clock source connected to the ADC.

Tutorial Contents

- Analog-to-Digital Conversion
- **Analog-to-Digital Converter (ADC) in ATmega328P**
  - **Features**
  - Associated Registers and Timing Diagram
- Lab 5 Validation

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Analog-to-Digital Converter (ADC) in ATmega328P

- ATmega328P has an on-chip Analog-to-Digital Converter (ADC). It is a 10-bit successive approximation ADC with 8 analog input channels. There are also 3 internal channels that can be selected with the multiplexer decoder. These are temperature sensor (channel 8), band-gap reference (1.1V) and GND (0V).

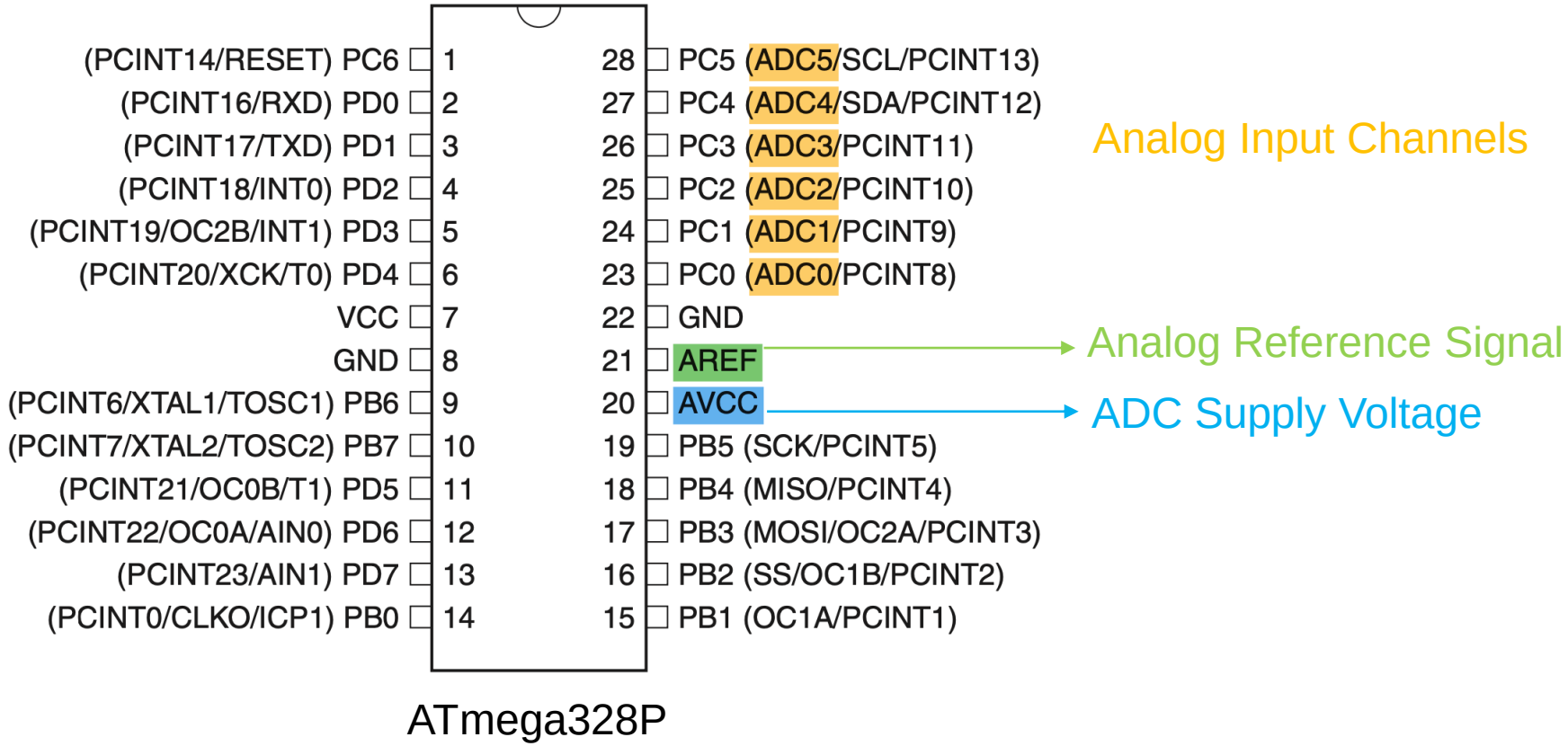
- It has a conversion time of $65\ \mu s$ to $260\ \mu s$.

- The ADC has a separate analog supply voltage pin, AV_{CC} which must not differ more than ±0.3V from V_{CC}.
- Internal reference voltages of nominally 1.1V or AV_{CC} are provided on-chip. The voltage reference may be externally decoupled at the AREF pin by a capacitor for better noise performance.

- The ADC converts an analog input voltage to a 10-bit digital value through successive approximation. The minimum value represents GND and the maximum value represents the voltage on the AREF pin minus 1 LSB (Least Significant Bit).

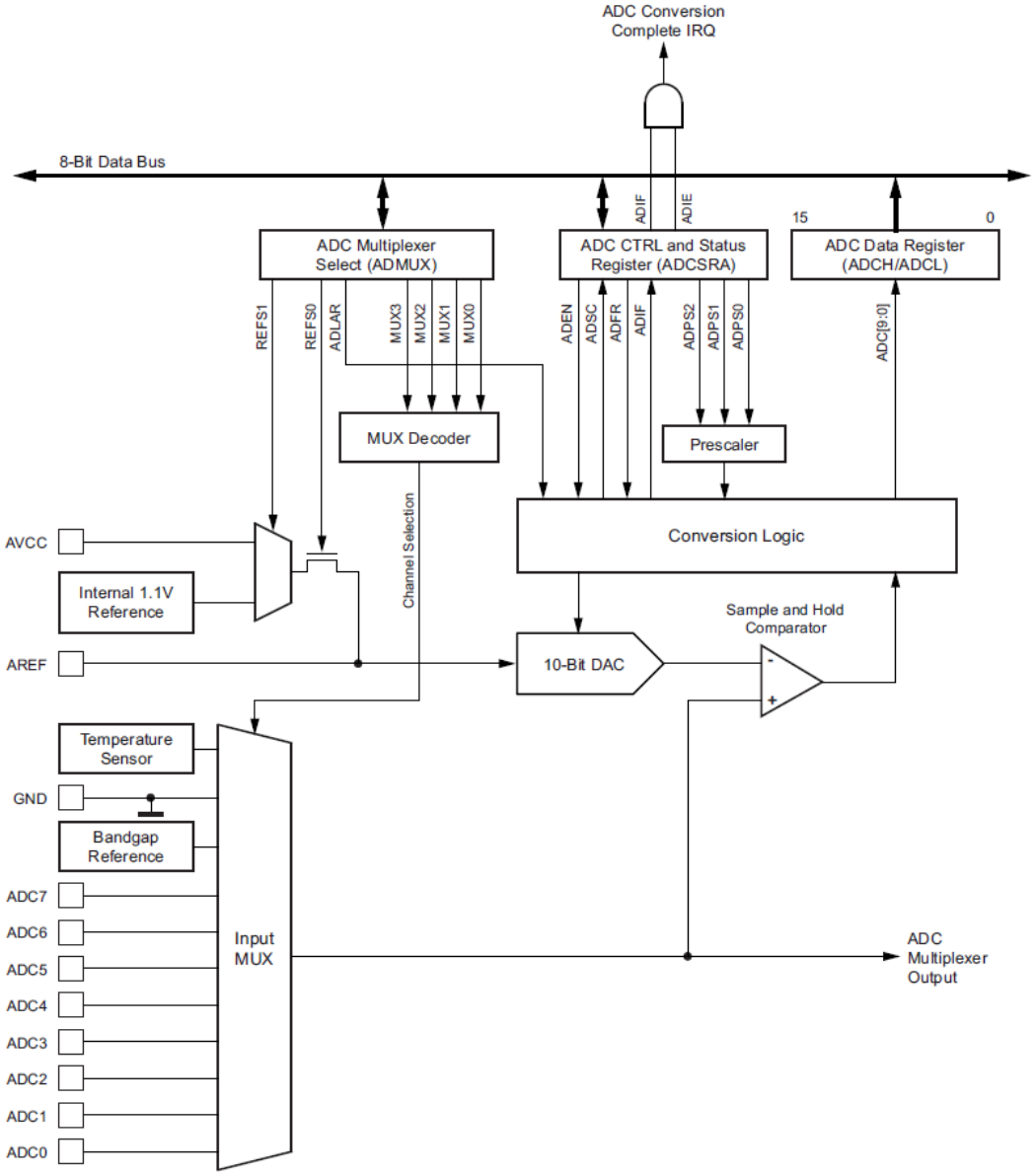
- ADC can be set up for free running conversion, single conversion, and interrupt based conversion.

Analog-to-Digital Converter (ADC) in ATmega328P



Analog-to-Digital Converter (ADC) in ATmega328P

Figure 23-1. Analog to Digital Converter Block Schematic Operation



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Tutorial Contents

- Analog-to-Digital Conversion
- **Analog-to-Digital Converter (ADC) in ATmega328P**
  - Features
  - **Associated Registers and Timing Diagram**
- Lab 5 Validation

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Analog-to-Digital Converter (ADC) in ATmega328P

- The ATmega328P on-chip ADC has the following associated registers:

ADMUX – ADC Multiplexer Selection Register

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:6 – REFS1:0: Reference Selection Bits**

These bits select the voltage reference for the ADC, as shown in Table 23-3. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set). The internal voltage reference options may not be used if an external reference voltage is being applied to the AREF pin.

Table 23-3. Voltage Reference Selections for ADC

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, internal V_{REF} turned off
0	1	AV_{CC} with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 1.1V voltage reference with external capacitor at AREF pin

- **Bit 5 – ADLAR: ADC Left Adjust Result**

The ADLAR bit affects the presentation of the ADC conversion result in the ADC data register. Write one to ADLAR to left adjust the result. Otherwise, the result is right adjusted. Changing the ADLAR bit will affect the ADC data register immediately, regardless of any ongoing conversions. For a complete description of this bit, see Section 23.9.3 "ADCL and ADCH – The ADC Data Register" on page 219.

Analog-to-Digital Converter (ADC) in ATmega328P

- The ATmega328P on-chip ADC has the following associated registers:

ADMUX – ADC Multiplexer Selection Register

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 3:0 – MUX3:0: Analog Channel Selection Bits**

The value of these bits selects which analog inputs are connected to the ADC. See Table 23-4 on page 218 for details. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set).

MUX3-MUX0	Single-Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6
0111	ADC7

MUX3-MUX0	Single-Ended Input
1000	Temperature Sensor
1001	(reserved)
1010	(reserved)
1011	(reserved)
1100	(reserved)
1101	(reserved)
1110	1.1 Volts (V_{BG})
1111	0 Volts (GND)

Analog-to-Digital Converter (ADC) in ATmega328P

- The ATmega328P on-chip ADC has the following associated registers:

ADCL and ADCH – The ADC Data Register

ADLAR = 0

Bit	15	14	13	12	11	10	9	8	
(0x79)	–	–	–	–	–	–	ADC9	ADC8	ADCH
(0x78)	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

ADLAR = 1

Bit	15	14	13	12	11	10	9	8	
(0x79)	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
(0x78)	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Analog-to-Digital Converter (ADC) in ATmega328P

- The ATmega328P on-chip ADC has the following associated registers:

ADCSRA – ADC Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	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	

- **Bit 7 – ADEN: ADC Enable**

Writing this bit to one enables the ADC. By writing it to zero, the ADC is turned off. Turning the ADC off while a conversion is in progress, will terminate this conversion.

- **Bit 6 – ADSC: ADC Start Conversion**

In single conversion mode, write this bit to one to start each conversion. In free running mode, write this bit to one to start the first conversion. The first conversion after ADSC has been written after the ADC has been enabled, or if ADSC is written at the same time as the ADC is enabled, will take 25 ADC clock cycles instead of the normal 13. This first conversion performs initialization of the ADC.

ADSC will read as one as long as a conversion is in progress. When the conversion is complete, it returns to zero. Writing zero to this bit has no effect.

Analog-to-Digital Converter (ADC) in ATmega328P

- The ATmega328P on-chip ADC has the following associated registers:

ADCSRA – ADC Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	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	

- **Bit 5 – ADATE: ADC Auto Trigger Enable**

When this bit is written to one, auto triggering of the ADC is enabled. The ADC will start a conversion on a positive edge of the selected trigger signal. The trigger source is selected by setting the ADC trigger select bits, ADTS in ADCSRB.

- **Bit 4 – ADIF: ADC Interrupt Flag**

This bit is set when an ADC conversion completes and the data registers are updated. The ADC conversion complete interrupt is executed if the ADIE bit and the I-bit in SREG are set. ADIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ADIF is cleared by writing a logical one to the flag. Beware that if doing a read-modify-write on ADCSRA, a pending interrupt can be disabled. This also applies if the SBI and CBI instructions are used.

- **Bit 3 – ADIE: ADC Interrupt Enable**

When this bit is written to one and the I-bit in SREG is set, the ADC conversion complete interrupt is activated.

Analog-to-Digital Converter (ADC) in ATmega328P

- The ATmega328P on-chip ADC has the following associated registers:

ADCSRA – ADC Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	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	

- **Bits 2:0 – ADPS2:0: ADC Prescaler Select Bits**

These bits determine the division factor between the system clock frequency and the input clock to the ADC.

ADPS2-ADPS0	Division Factor
000	2
001	2
010	4
011	8
100	16
101	32
110	64
111	128

By default, the successive approximation circuitry requires an input clock frequency between 50kHz and 200kHz to get maximum resolution.

Analog-to-Digital Converter (ADC) in ATmega328P

- The ATmega328P on-chip ADC has the following associated registers:

ADCSRB – ADC Control and Status Register B

Bit	7	6	5	4	3	2	1	0	
(0x7B)	–	ACME	–	–	–	ADTS2	ADTS1	ADTS0	ADCSRB
Read/Write	R	R/W	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 2:0 – ADTS2:0: ADC Auto Trigger Source**

If ADATE in ADCSRA is written to one, the value of these bits selects which source will trigger an ADC conversion. If ADATE is cleared, the ADTS2:0 settings will have no effect. A conversion will be triggered by the rising edge of the selected interrupt flag. Note that switching from a trigger source that is cleared to a trigger source that is set, will generate a positive edge on the trigger signal. If ADEN in ADCSRA is set, this will start a conversion. Switching to free running mode (ADTS[2:0]=0) will not cause a trigger event, even if the ADC interrupt flag is set.

Table 23-6. ADC Auto Trigger Source Selections

ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free running mode
0	0	1	Analog comparator
0	1	0	External interrupt request 0
0	1	1	Timer/Counter0 compare match A
1	0	0	Timer/Counter0 overflow
1	0	1	Timer/Counter1 compare match B
1	1	0	Timer/Counter1 overflow
1	1	1	Timer/Counter1 capture event

Analog-to-Digital Converter (ADC) in ATmega328P

- The ATmega328P on-chip ADC has the following associated registers:

DIDR0 – Digital Input Disable Register 0

Bit	7	6	5	4	3	2	1	0	
(0x7E)	–	–	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	DIDR0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 5:0 – ADC5D..ADC0D: ADC5..0 Digital Input Disable**

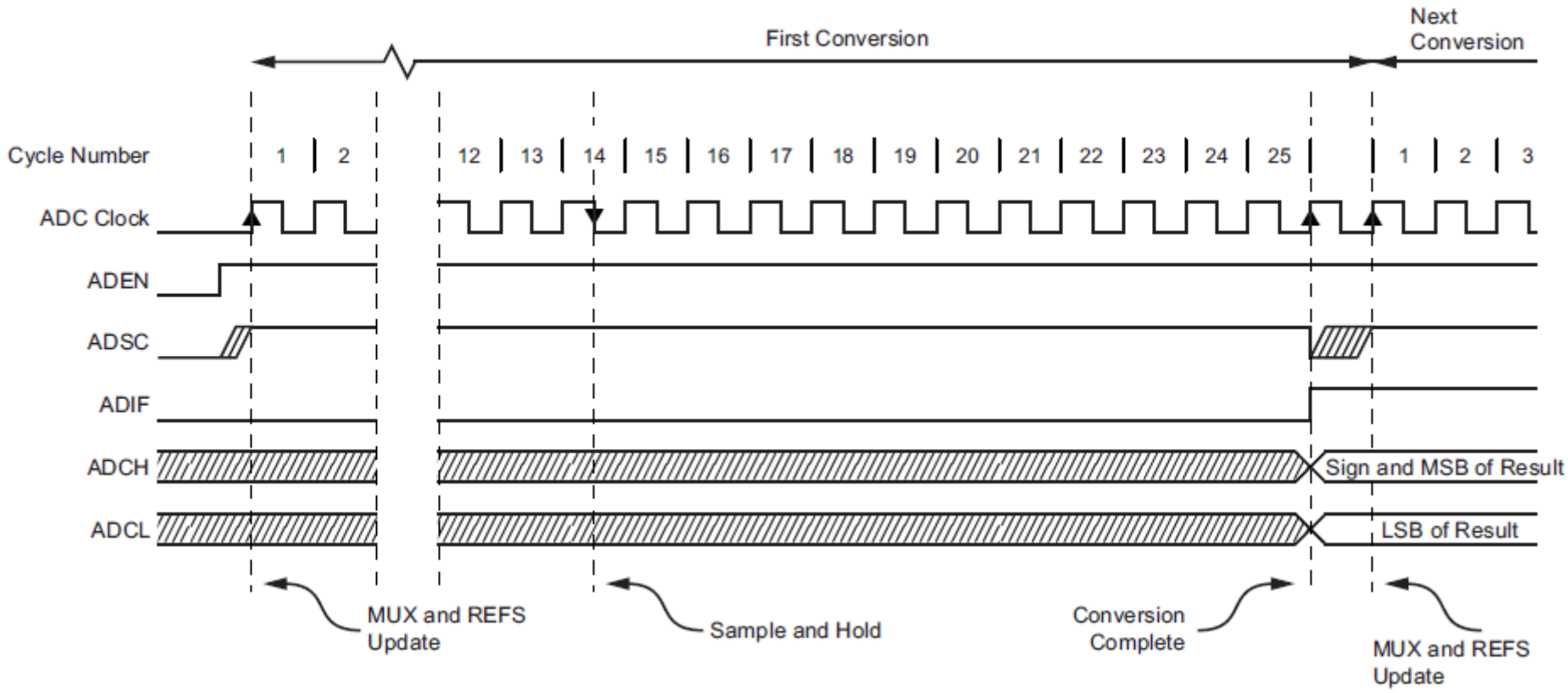
When this bit is written logic one, the digital input buffer on the corresponding ADC pin is disabled. The corresponding PIN register bit will always read as zero when this bit is set. When an analog signal is applied to the ADC5..0 pin and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.

Note that ADC pins ADC7 and ADC6 do not have digital input buffers, and therefore do not require digital input disable bits.

Analog-to-Digital Converter (ADC) in ATmega328P

- Here is the timing diagram for the ADC conversion:

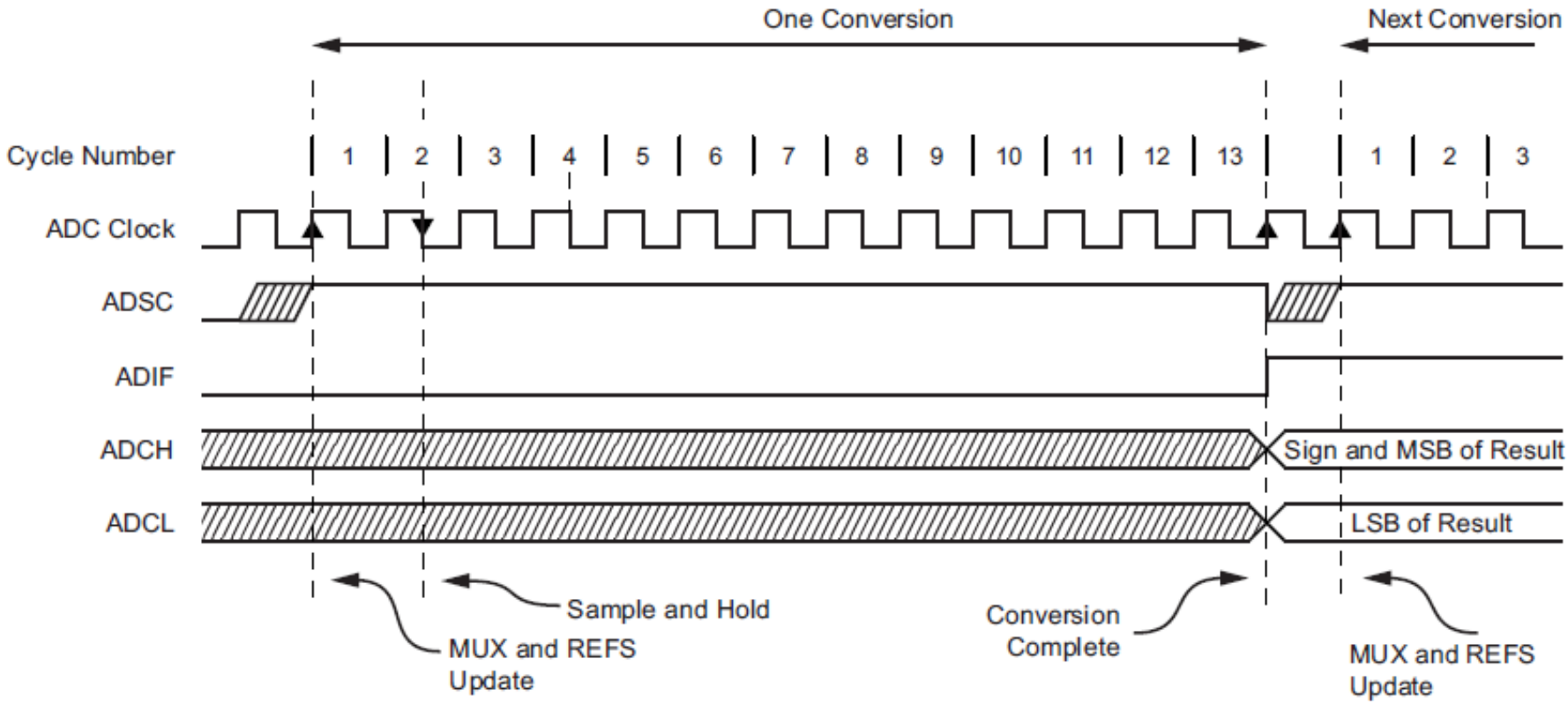
Figure 23-4. ADC Timing Diagram, First Conversion (Single Conversion Mode)



Analog-to-Digital Converter (ADC) in ATmega328P

- Here is the timing diagram for the ADC conversion:

Figure 23-5. ADC Timing Diagram, Single Conversion



Analog-to-Digital Converter (ADC) in ATmega328P

- Here is the timing diagram for the ADC conversion:

Figure 23-7. ADC Timing Diagram, Free Running Conversion

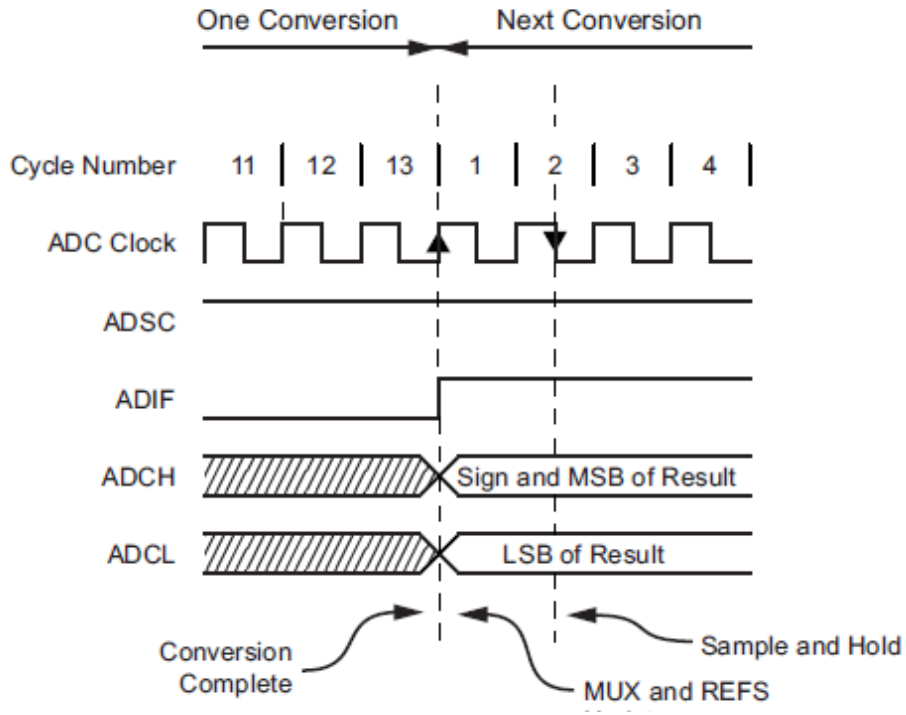


Table 23-1. ADC Conversion Time

Condition	Sample and Hold (Cycles from Start of Conversion)	Conversion Time (Cycles)
First conversion	13.5	25
Normal conversions, single ended	1.5	13
Auto triggered conversions	2	13.5

Tutorial Contents

- Analog-to-Digital Conversion
- Analog-to-Digital Converter (ADC) in ATmega328P
  - Features
  - Associated Registers and Timing Diagram
- **Lab 5 Validation**

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Analog-to-Digital Converter (ADC) in ATmega328P

Lab 5 Validation: (to be submitted in your Lab next week)

It is required to build a circuit with a potentiometer and 10 LED indicators. The potentiometer is a variable resistance which enables the user to change the voltage signal as desired.

Circuit Operation:

- When the user changes the potentiometer, the voltage reading is converted to a 10-bit digital word to be displayed on the 10 LED indicators.
- For example, when the potentiometer is on maximum resistance (i.e. zero input voltage), all 10 LED indicators are off. When it is on minimum resistance (i.e. maximum voltage), all 10 LED indicators are on.

Design Requirements:

- Use the AVR embedded board.
- Use Embedded C for programming.
- The 10 LED indicators are connected to PD7-PD0 and PB1-PB0.
- The potentiometer is connected to ADC0.
- Include a reset push button in your hardware implementation.
- Do not use pre-defined ADC-related functions.

Analog-to-Digital Converter (ADC) in ATmega328P

Lab 5 Validation: (to be submitted in your Lab next week)

Here is the pseudo-code of the software approach to be used.

1. Input-Output Configuration:
    a) Configure PD7-PD0 and PB1-PB0 as outputs.

2. ADC Initialization:
    a) Select reference voltage source to be external connected to AREF. The default values of REFS0 and REFS1 already do so.
    b) Configure the ADC result to be right aligned. The default value of ADLAR already does that.
    c) Choose a proper ADC pre-scaler so that the ADC clock frequency is between 50kHz and 200 kHz. So, choose a pre-scaler of 128 such that the ADC cock frequency will be $\frac{16\ MHz}{128} = 125\ kHz$. Thus, set the bits ADPS2-ADPS0 in ADCSRA to HIGH.
    d) Enable the ADC module by setting ADEN in ADCSRA to HIGH.
    e) Disable the digital input buffer on ADC0. Set ADC0D in DIDR0 to HIGH.

Analog-to-Digital Converter (ADC) in ATmega328P

Lab 5 Validation: (to be submitted in your Lab next week)

3. Reading Analog Input Through ADC:
    a) Configure the ADC multiplexer to choose ADC0. The default values of MUX3-MUX0 already do so.
    b) Start the ADC conversion by setting the ADSC bit in ADCSRA to HIGH. This way the ADC is running on single conversion mode.
    c) Wait until the ADSC bit in ADCSRA is set to LOW which means that the conversion ended. (Or you can also wait until the ADIF bit in ADCSRA is set to HIGH and it should be cleared in code.)
    d) Read the ADC output from ADCH and ADCL.

4. Output the ADC output to be displayed on the LED indicators on PD7-PD0 and PB1-PB0.

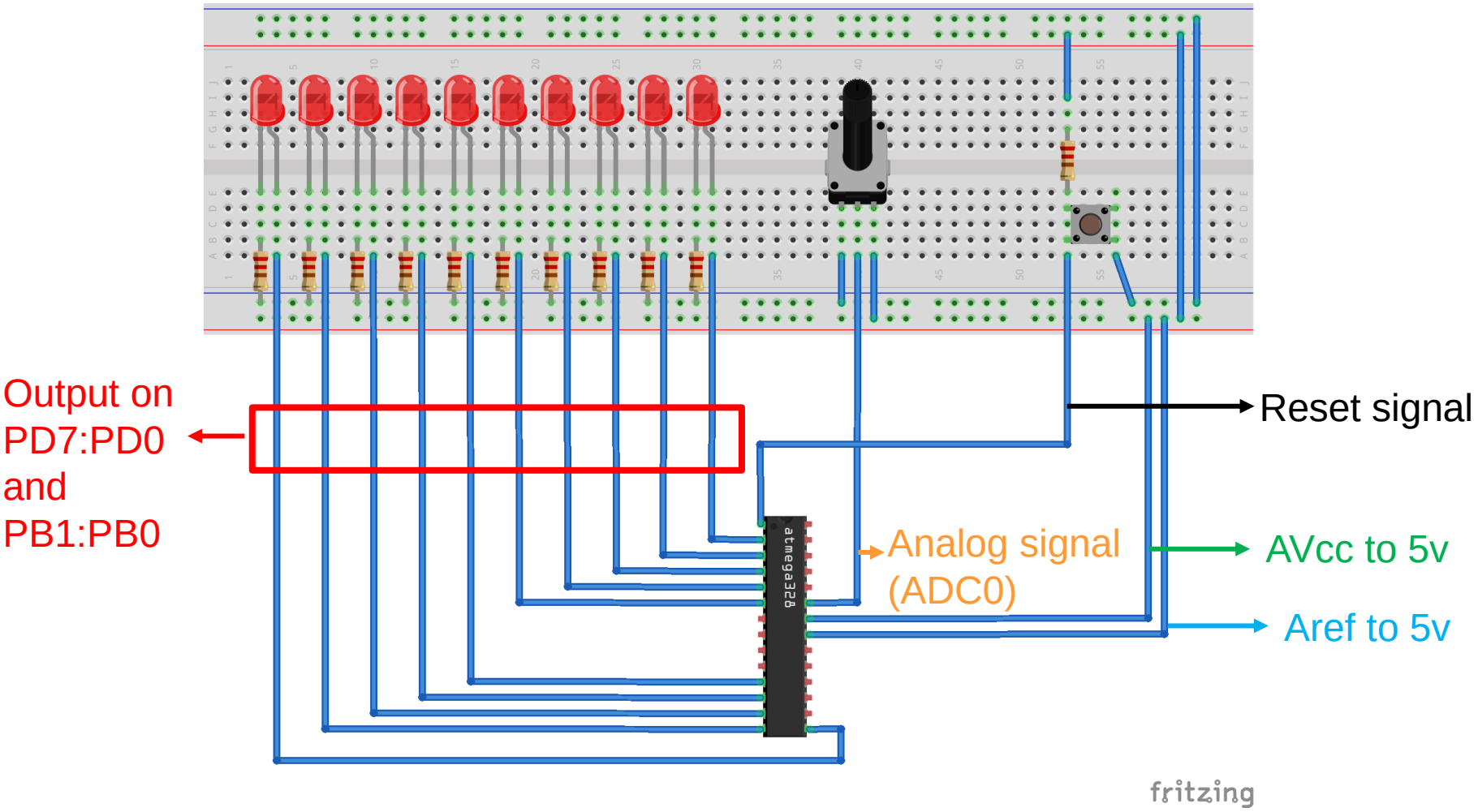
5. Go to step 3(b) and repeat.

Recommended: Run the ADC on free-running mode instead of single conversion mode and/or using interrupts.

Analog-to-Digital Converter (ADC) in ATmega328P

Lab 5 Validation: (to be submitted in your Lab next week)

Here are the hardware connections:



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science