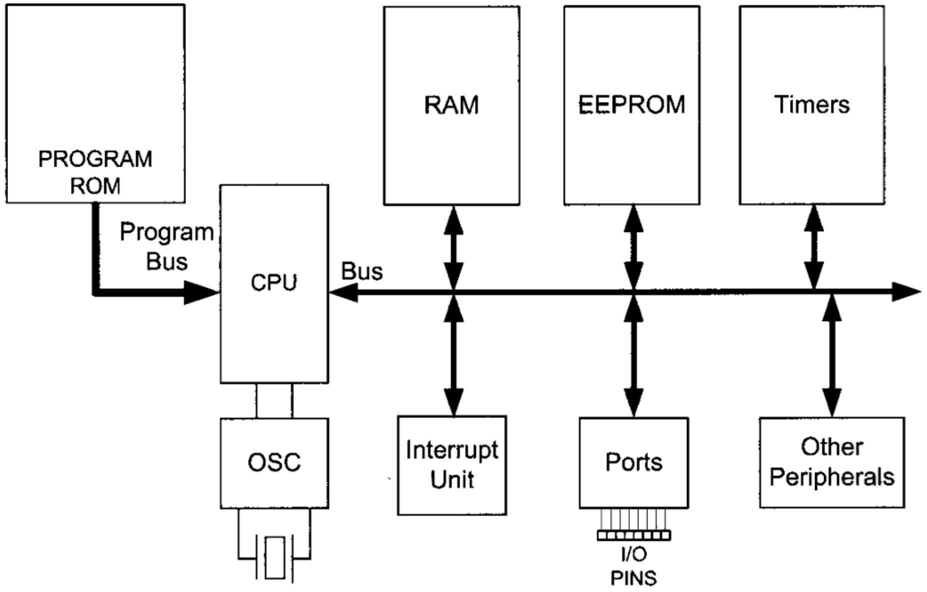
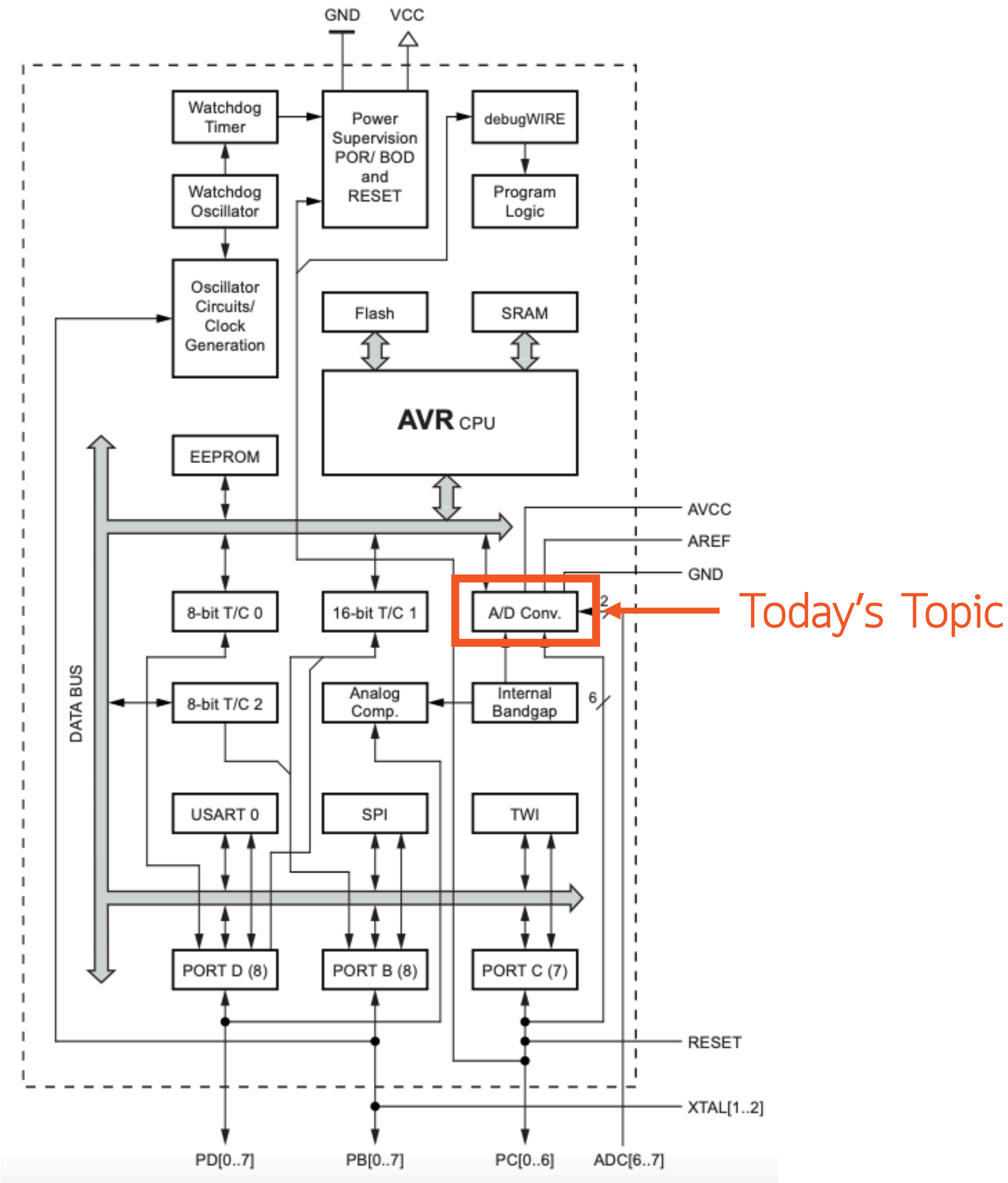
AVR Analog-to-Digital Convertor

Recall AVR Architecture



Simplified AVR Architecture



ATMega328P Architecture

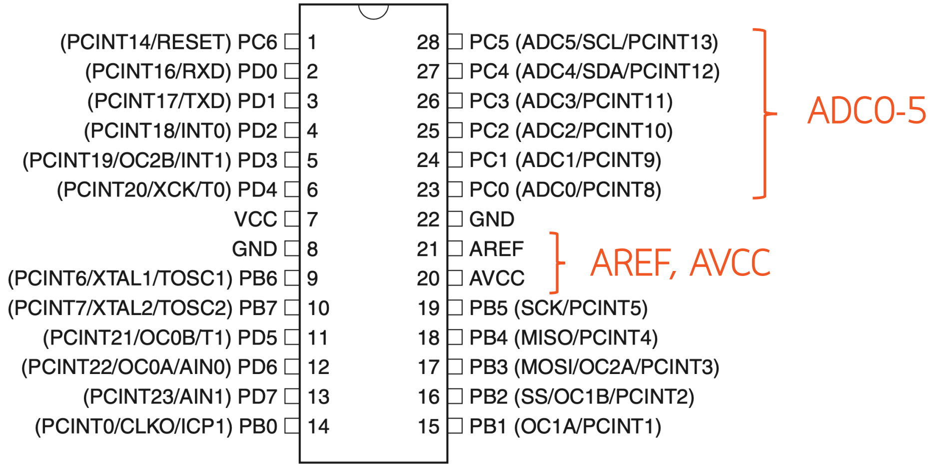
AVR* ADC: Features

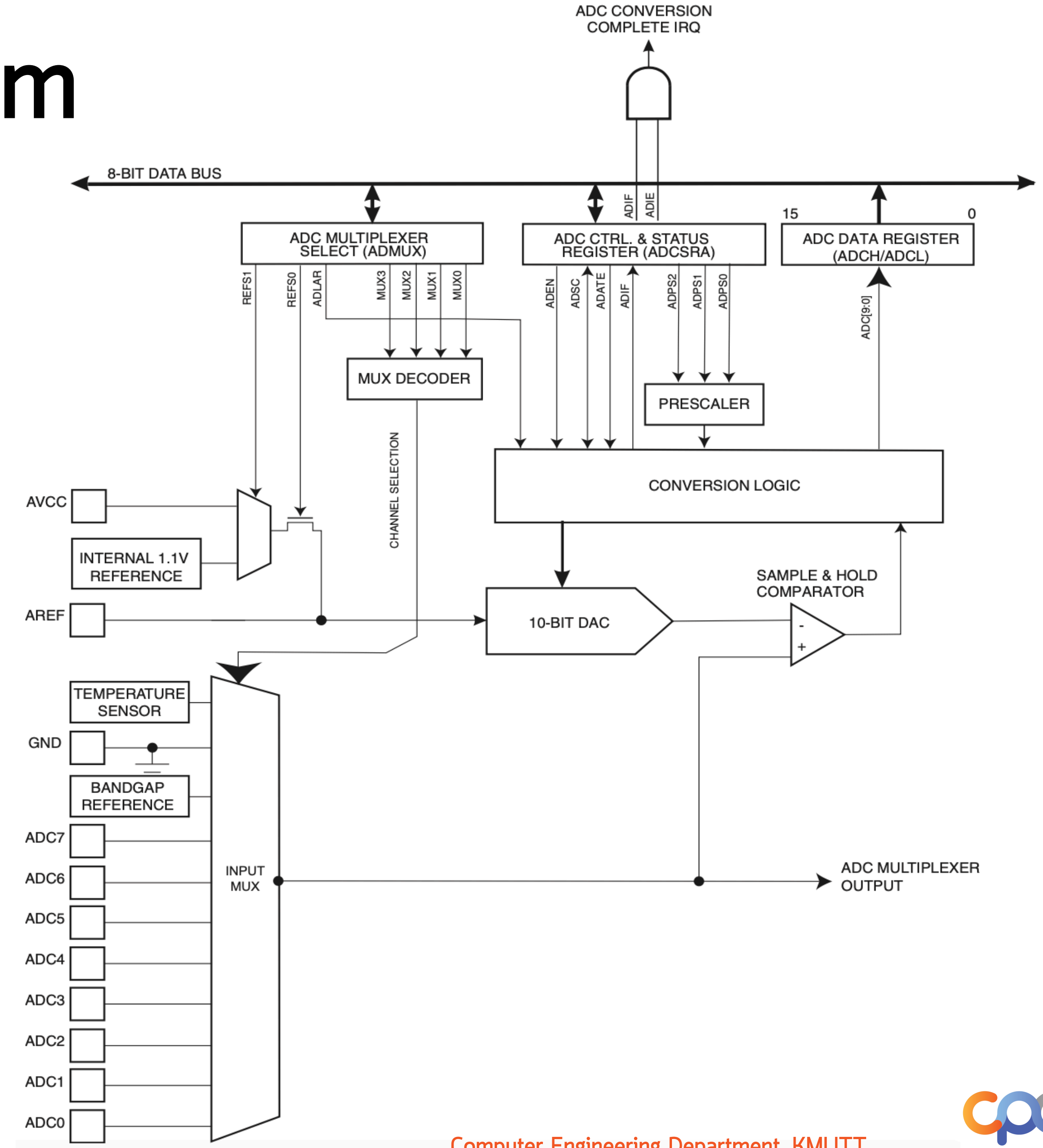
- 10-bit resolution, 8 multiplexed single ended input channels
- Up to 76.9kSPS (15kSPS at Maximum Resolution), 13 - 260us Conversion Time
- 0 - V_{CC} ADC input voltage range (selectable 1.1V ADC reference voltage)
- Free running or single conversion mode
- Interrupt on ADC conversion complete

* The information in this slide are based on ATmega328P. However, they are mostly applied to other AVR microcontrollers

AVR ADC: Block Diagram

- 4 control/data registers: ADMUX, ADCSRA, ADCH/L
- 8-10 I/O pins: AVCC, AREF, ADC0-ADC7





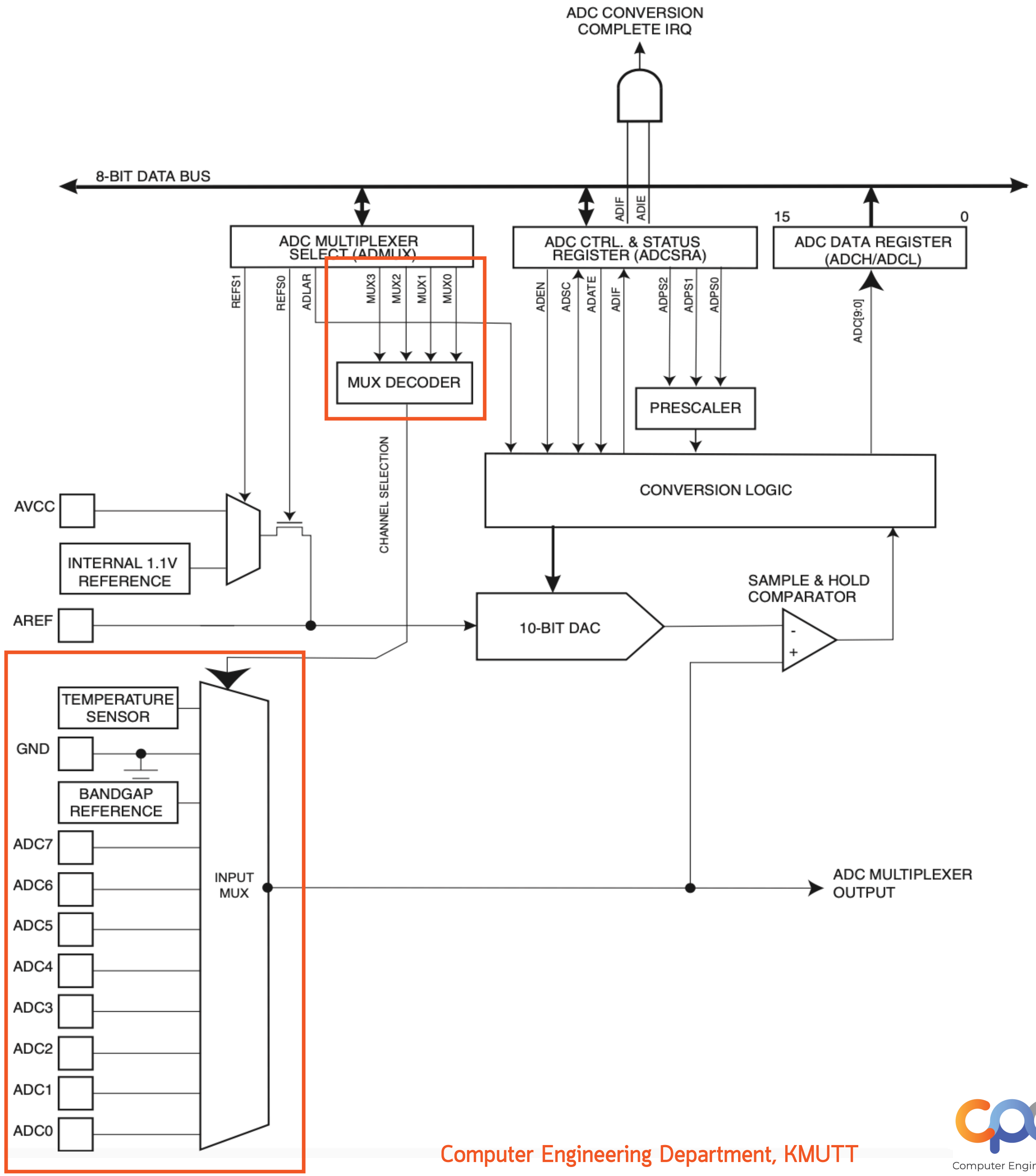
AVR ADC: ADMUX

- Analog input channel can be selected by writing to the MUX[3:0] bits in ADMUX

Table 24-4. Input Channel Selections

MUX3...0	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6
0111	ADC7
1000	ADC8[1]
1001	(reserved)
1010	(reserved)
1011	(reserved)
1100	(reserved)
1101	(reserved)
1110	1.1V (V_{BG})
1111	0V (GND)

Note: 1. For Temperature Sensor.

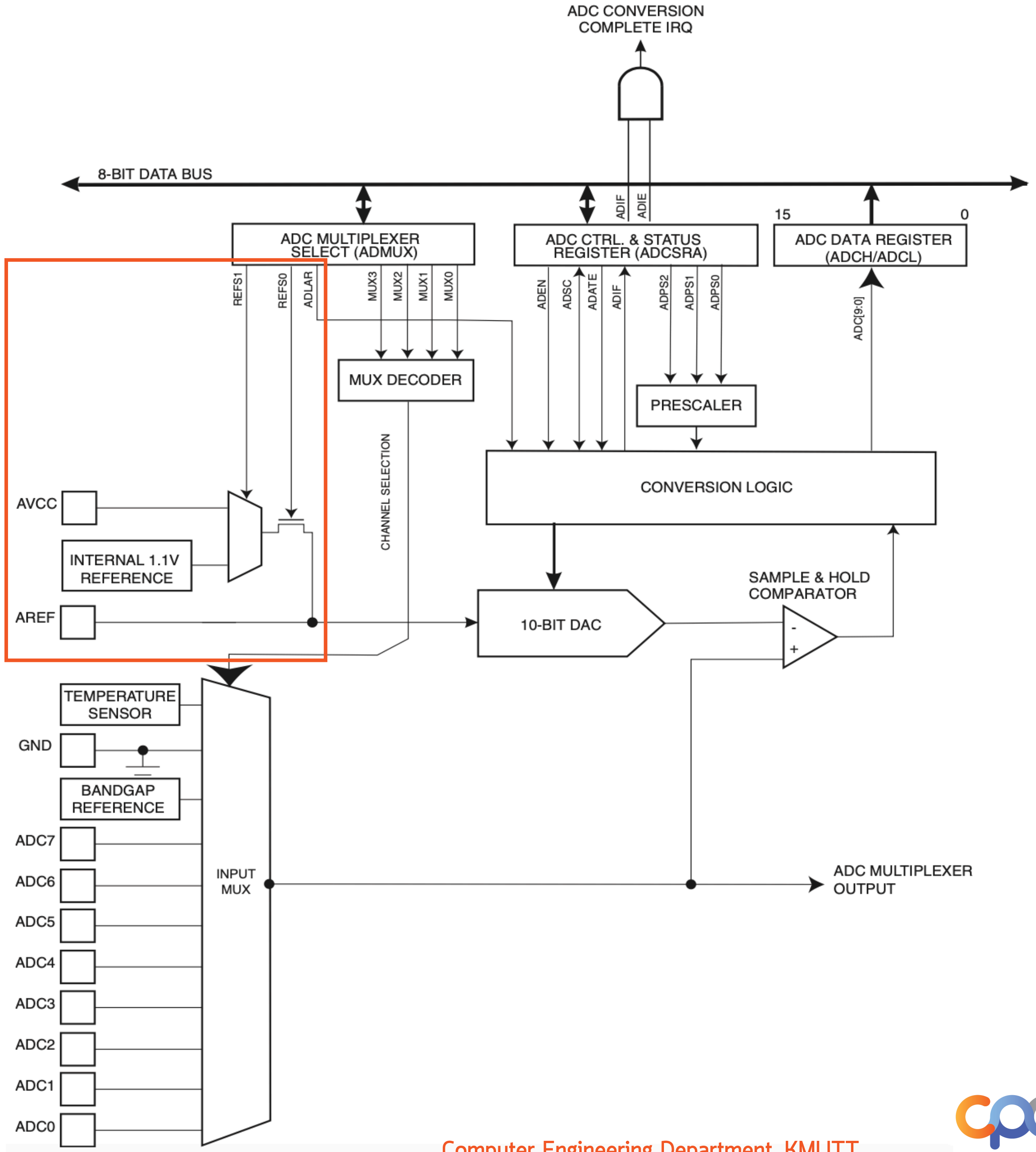


AVR ADC: ADMUX

- Voltage reference can be selected by writing to the REFS[1:0] bits in ADMUX
- Changes during conversion will not go in effect until this conversion is complete

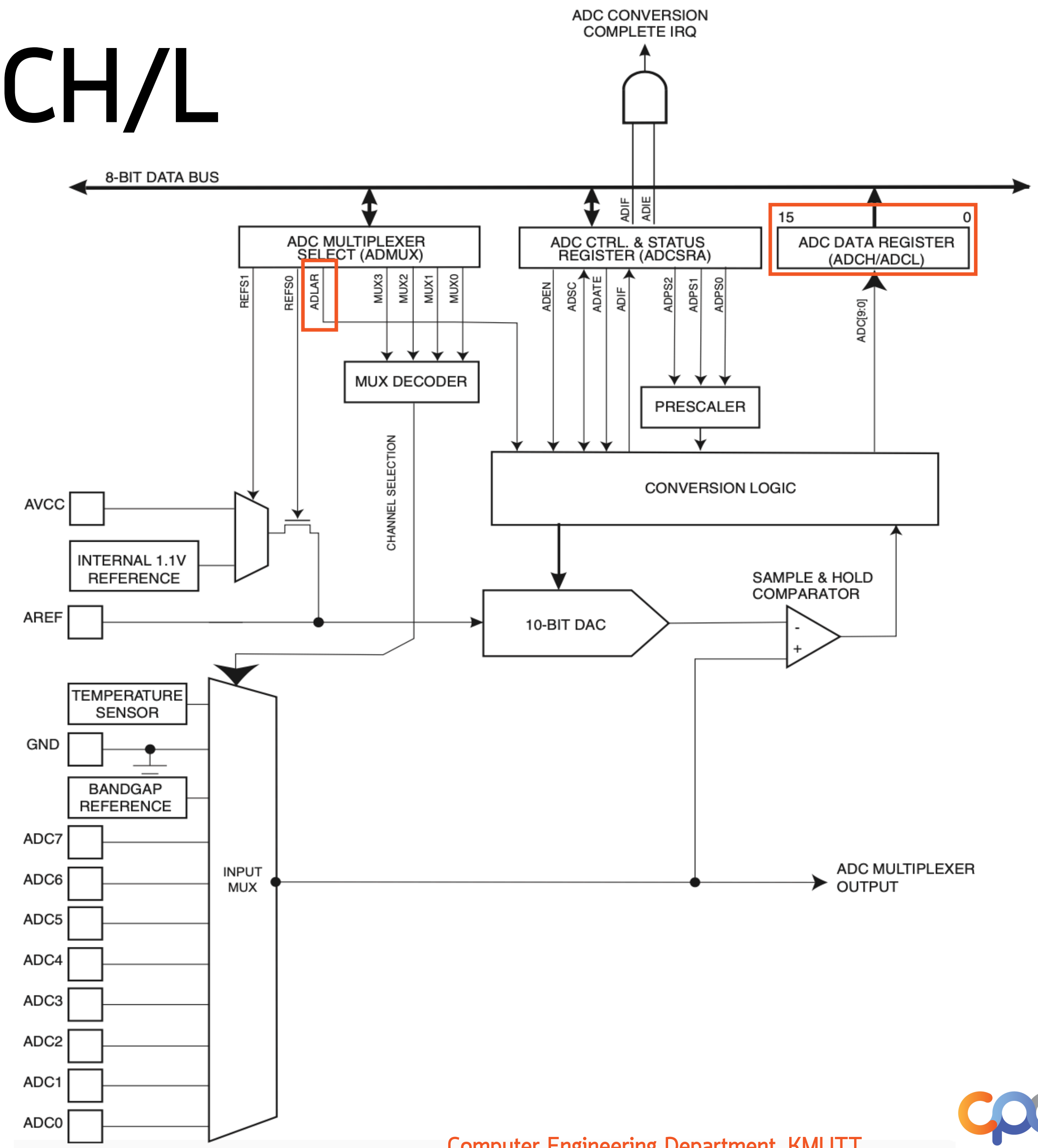
Table 24-3. **Voltage Reference Selections for ADC**

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



AVR ADC: ADMUX, ADCH/L

- ADC conversion result will be stored in ADC Data Registers, ADCH and ADCL
- The result is left adjusted when ADLAR = 1 and right adjusted when ADLAR = 0
- ADC data registers will be updated after ADCH is read (we may retrieve 8-bits value by reading ADCH and disregard ADCL when ADLAR = 1)



AVR ADC: ADMUX, ADCH/L

- ADC conversion result will be stored in ADC Data Registers, ADCH and ADCL
- The result is left adjusted when ADLAR = 1 and right adjusted when ADLAR = 0
- ADC data registers will be updated after ADCH is read (we may retrieve 8-bits value by reading ADCH and disregard ADCL when ADLAR = 1)

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	

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

Visit https://stackoverflow.com/questions/57534519/atmega328p-convert-analog-value-to-voltage to learn why we multiply by 1024 instead of 1023

AVR ADC: ADCSRA (Clock Source)

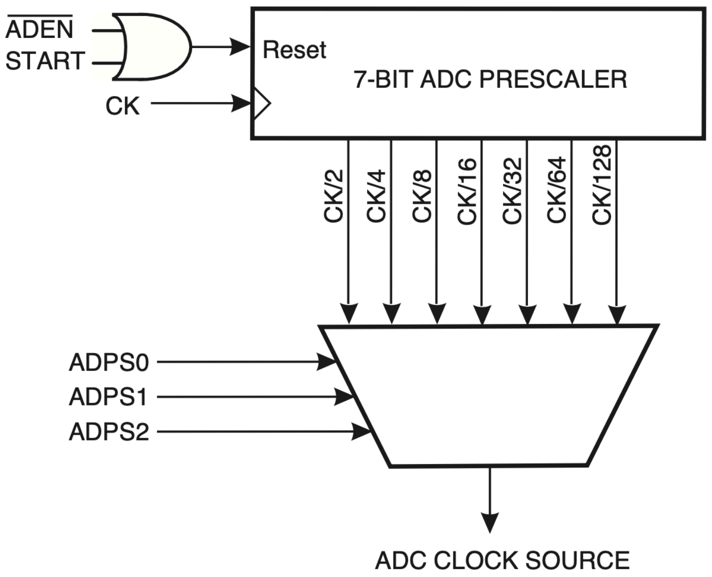


Table 24-5. ADC Prescaler Selections

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

- The ADC requires an input clock frequency between 50kHz and 200kHz to get maximum resolution
- ADC module contains a prescaler to generate ADC clock frequency from any CPU frequency. The prescaling is set by the ADPS[2:0] bits in ADCSRA

AVR ADC: ADCSRA/B (Starting a conversion)

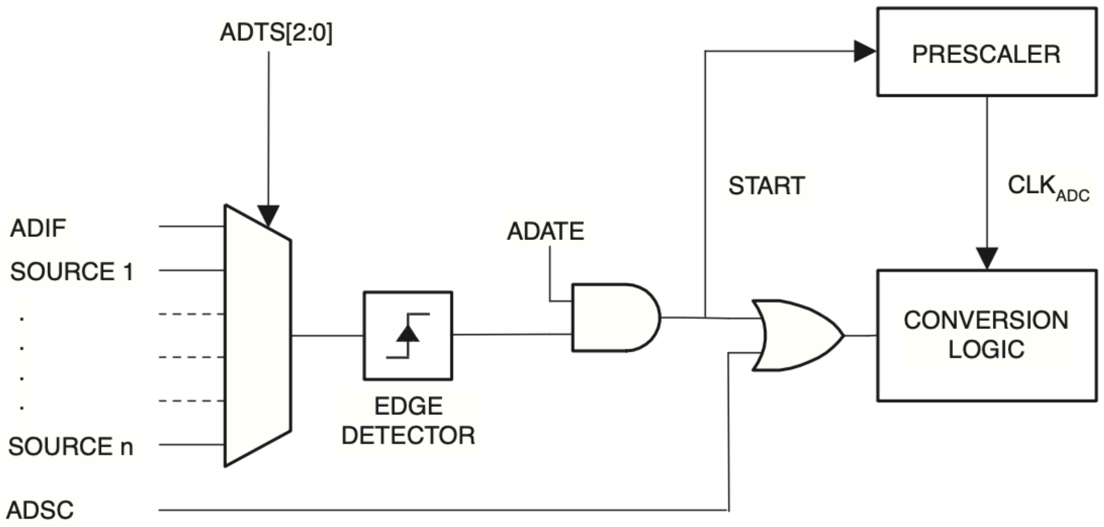


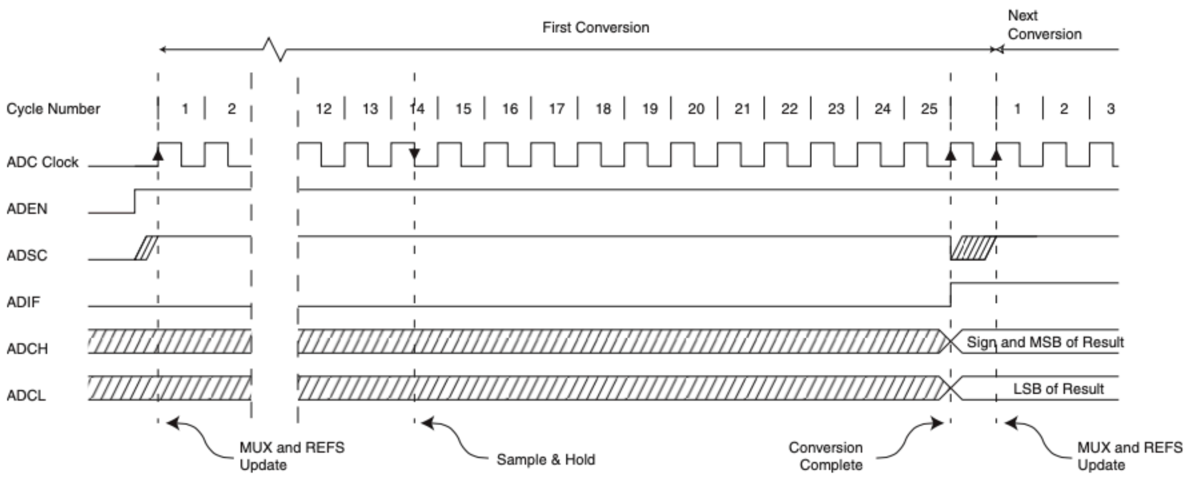
Table 24-6. ADC Auto Trigger Source Selections

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

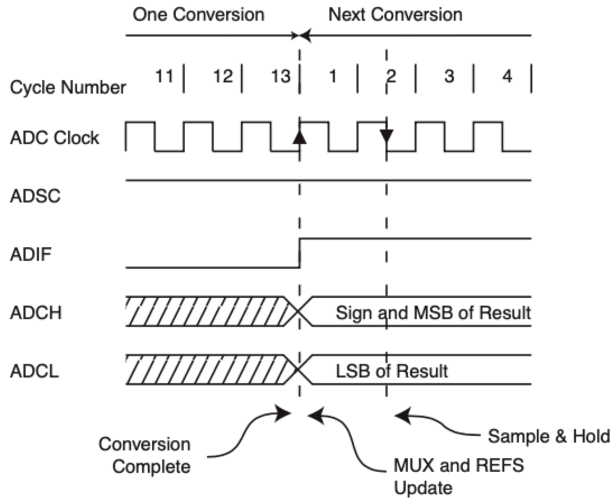
1. Single Conversion
   - set ADSC (ADC Start Conversion bit) to 1 to start each conversion
   - ADSC will stay 1 while a conversion is in progress and will be cleared by hardware when it is completed
2. Auto Trigger
   - set ADATE (ADC Auto Trigger Enable bit) to 1 to enable auto triggering mode
   - ADTS (ADC Trigger Select bits) in ADCSRB can be used to select a trigger source

AVR ADC: Timing

ADC Timing Diagram, First Conversion (Single Conversion Mode)



ADC Timing Diagram, Free Running Conversion



ADC Timing Diagram, Single Conversion

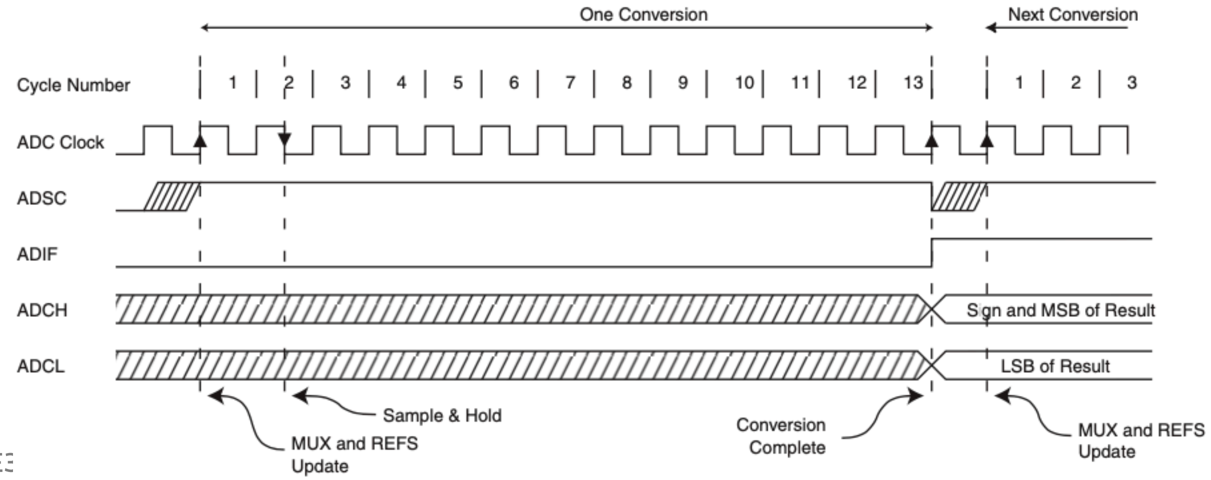
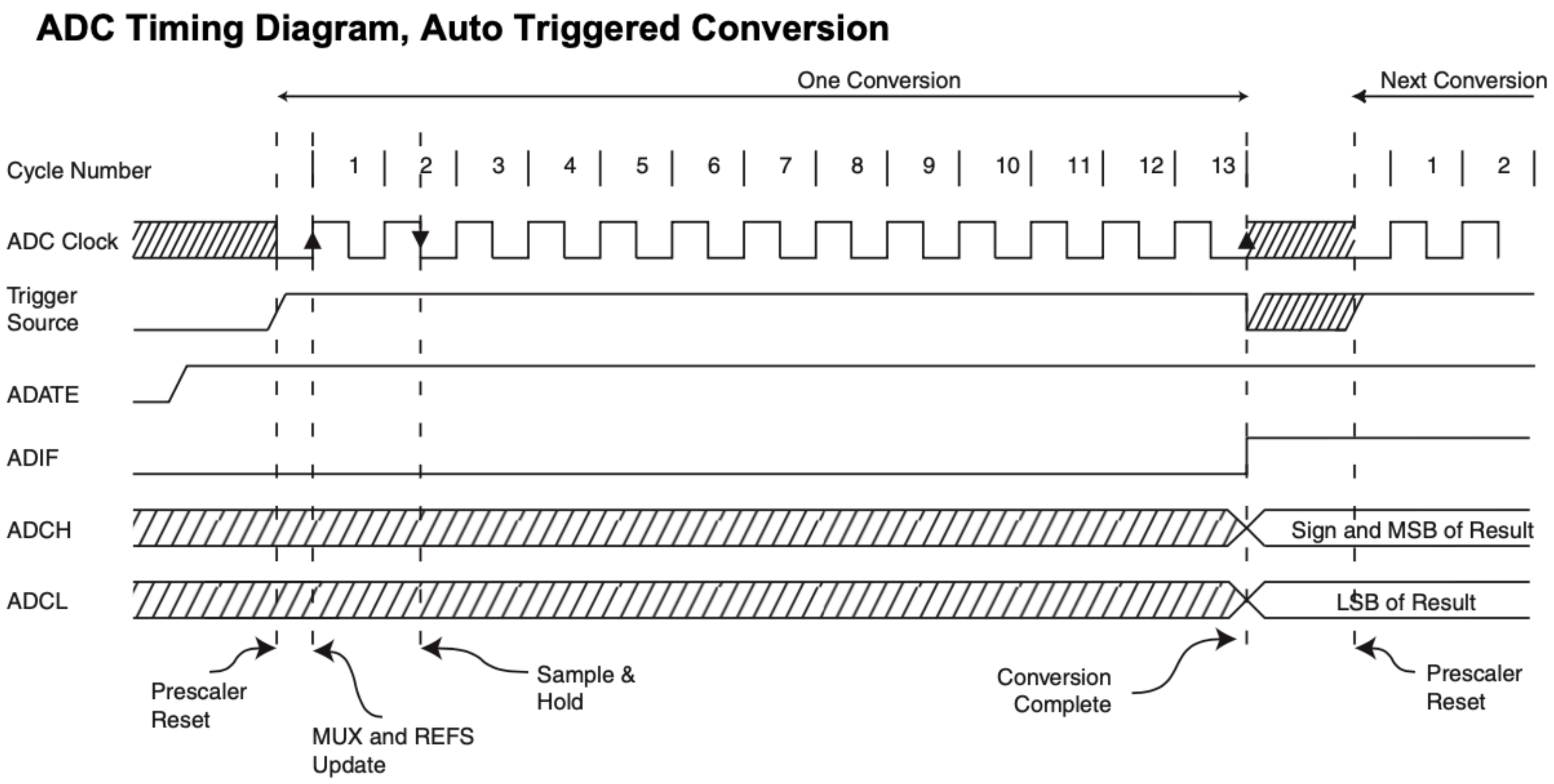


Table 24-1. ADC Conversion Time

Condition	Sample & 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

AVR ADC: Timing



New conversion will not be started if the trigger signal is still set after the conversion complete or if another positive edge occurs during conversion.

AVR ADC: Power Saving Consideration

PRR – Power Reduction Register

Bit	7	6	5	4	3	2	1	0	
(0x64)	PRTWI	PRTIM2	PRTIM0	–	PRTIM1	PRSPI	PRUSART0	PRADC	PRR
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	

Write PRADC to 1 to shutdown the ADC. The ADC must be disabled before shut down. The analog comparator cannot use the ADC input MUX when the ADC is shut down.

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	

Write these bit to logic 1 to disable the digital input buffer on the corresponding ADC pin when the digital input from this pin is not needed

Lab 5: AVR ADC Programming

1. Connect a circuit with an ATMega328P and MCP9700A (analog temperature sensor) and write a program to transmit the current ambient temperature to your PC using a serial port
2. Extend your program to display the current ambient temperature on the 16x2 alphanumeric LCD display in addition to transmiting through the serial port