CSE 315
Microprocessors & Microcontrollers

Tanvir Ahmed Khan

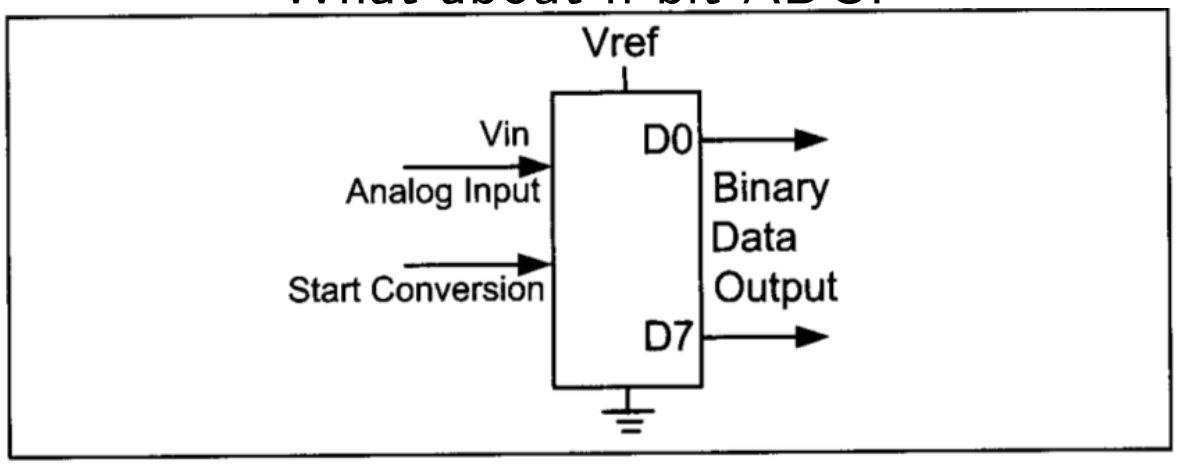
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology.

October 21, 2014

Recap

ADC Basics

Typical Digital System Characteristics
1-bit ADC
What about n-bit ADC?



ADC Jargons

- Sampling
- Quantization
- Resolution/Step Size
- Conversion Time
- V_{ref}
- Digital Data Output

Resolution/Step Size

- for n-bit ADC,
  - $stepsize = \frac{V_{ref}}{2^n}$
- Find the step size if,
  - $n = 8, V_{ref} = 2.56V$
  - $n = 10, V_{ref} = 2.56V$

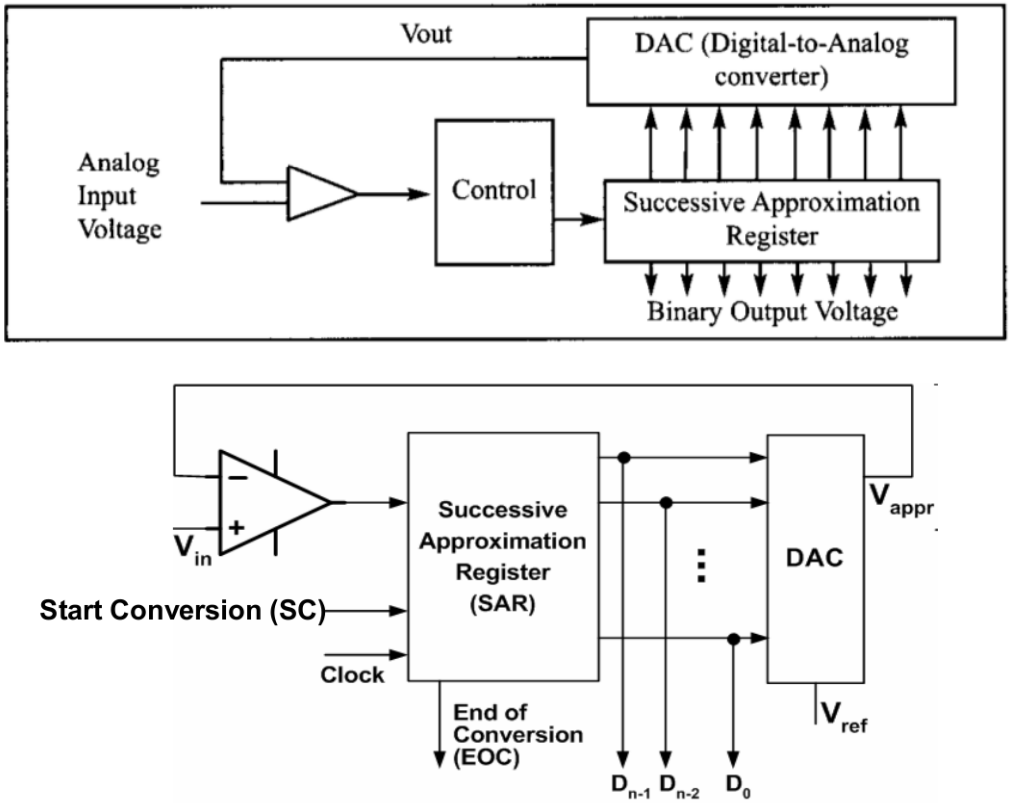
Digital Data Output

- $D_{out} = \left\lfloor \frac{V_{in}}{stepsize} \right\rfloor$
- For, $n = 9$ and $V_{ref} = 2.56V$, find the D_{out} if,
  - $V_{in} = 0.8V$
  - $V_{in} = 2.1V$

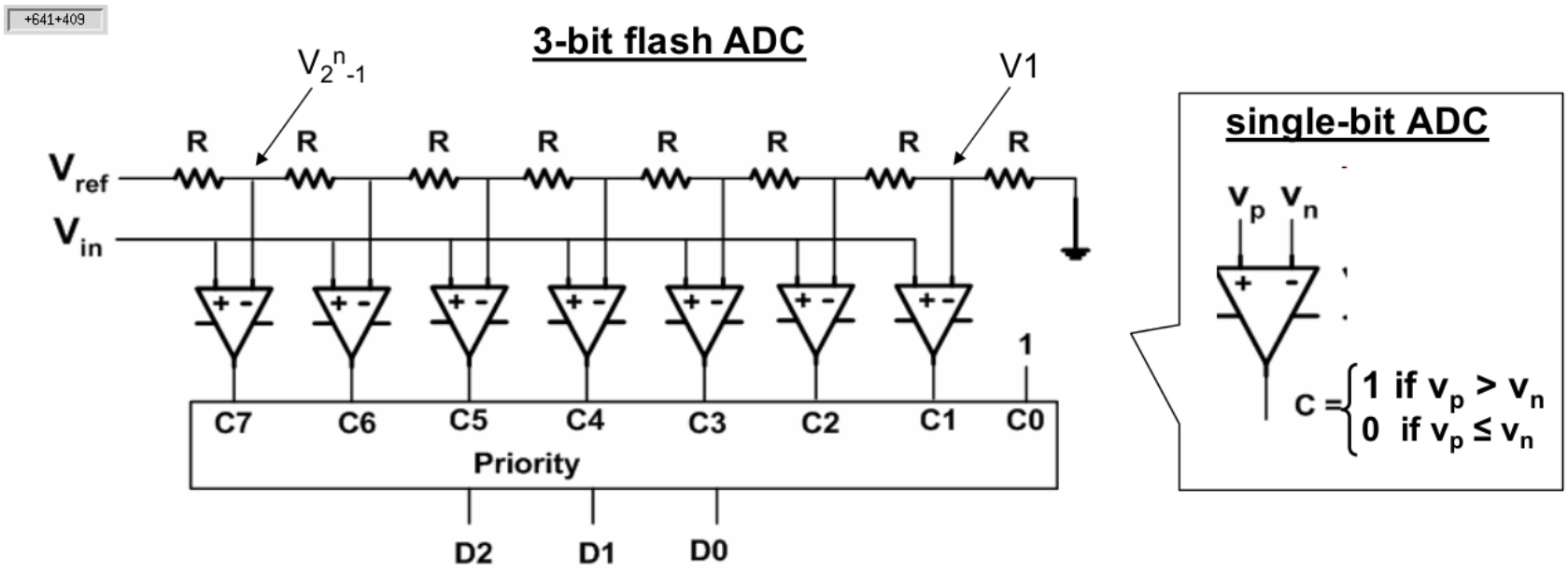
Analog-to-Digital Conversion Technologies

- Successive Approximation
- Integration
- Counter Based Conversion
- Parallel Conversion
  - Flash ADC

Successive Approximation ADC



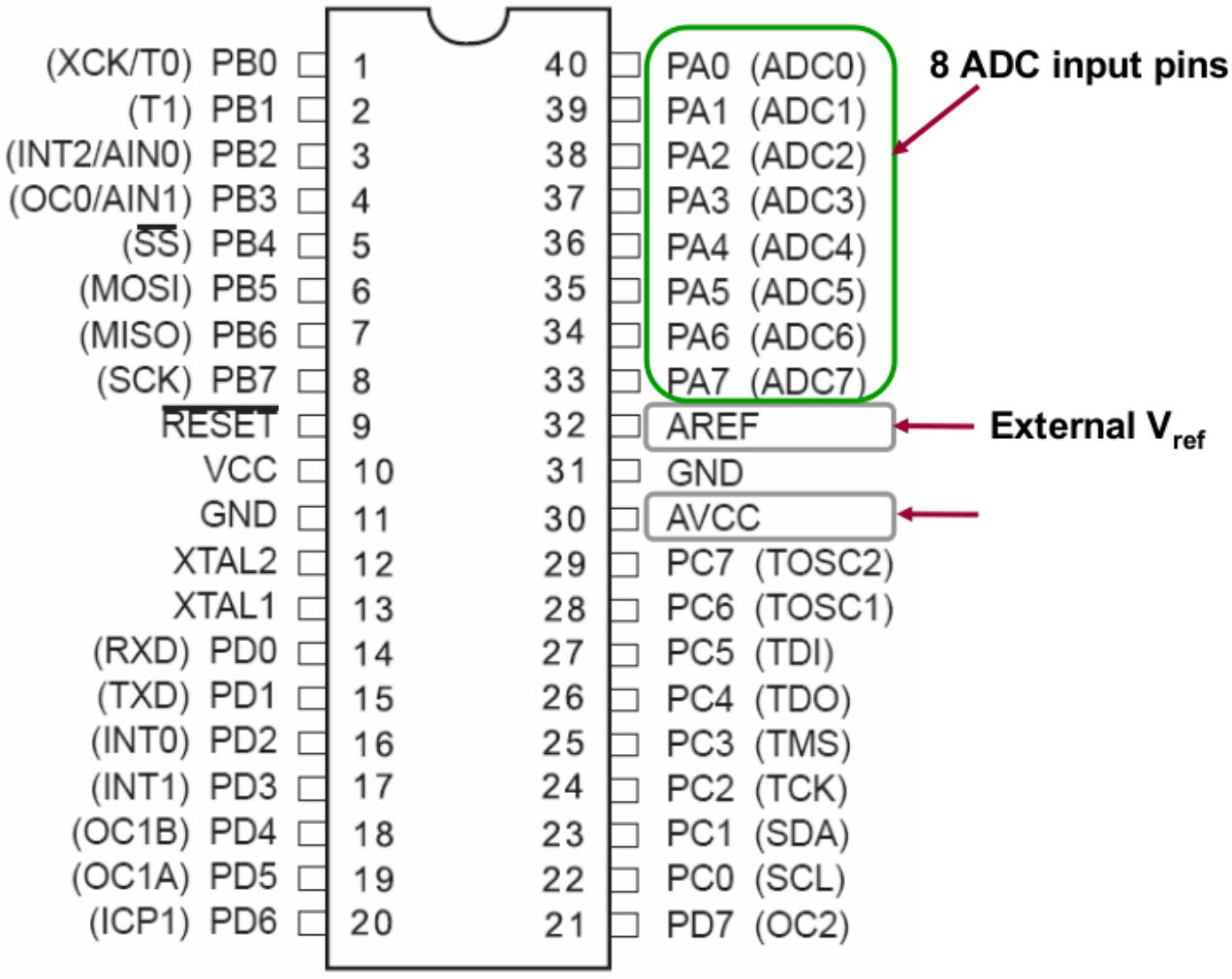
Flash ADC


+641+409
3-bit flash ADC
V_{2^n-1}
V1
R
V_{ref}
V_{in}
1
C7
C6
C5
C4
C3
C2
C1
C0
Priority
D2
D1
D0
single-bit ADC
v_p
v_n
C =
1 if $v_p > v_n$
0 if $v_p \leq v_n$

Today's Topic

ADC Programming in C

ATmega16/32 ADC Relevant Pin Diagram



ATmega16/32 ADC Features

ATmega16/32 ADC Features

- 10-bit ADC

ATmega16/32 ADC Features

- 10-bit ADC
- 2 output registers,

ATmega16/32 ADC Features

- 10-bit ADC
- 2 output registers,
  - `ADCH:ADCL`

ATmega16/32 ADC Features

- 10-bit ADC
- 2 output registers,
  - `ADCH:ADCL`
  - 16-bits, 6-bits are unused

ATmega16/32 ADC Features

- 10-bit ADC
- 2 output registers,
  - `ADCH:ADCL`
  - 16-bits, 6-bits are unused
  - Option to adjust left or right

ATmega16/32 ADC Features

- 10-bit ADC
- 2 output registers,
  - `ADCH:ADCL`
  - 16-bits, 6-bits are unused
  - Option to adjust left or right
- 8 Analog input channels,

ATmega16/32 ADC Features

- 10-bit ADC
- 2 output registers,
  - `ADCH:ADCL`
  - 16-bits, 6-bits are unused
  - Option to adjust left or right
- 8 Analog input channels,
  - 7 differential input

ATmega16/32 ADC Features

- 10-bit ADC
- 2 output registers,
  - `ADCH:ADCL`
  - 16-bits, 6-bits are unused
  - Option to adjust left or right
- 8 Analog input channels,
  - 7 differential input
  - with optional gain of 10x & 200x

ATmega16/32 ADC Features

- 10-bit ADC
- 2 output registers,
  - `ADCH:ADCL`
  - 16-bits, 6-bits are unused
  - Option to adjust left or right
- 8 Analog input channels,
  - 7 differential input
  - with optional gain of 10x & 200x
  - However, only one conversion at a time

ATmega16/32 ADC Features

- 10-bit ADC
- 2 output registers,
  - `ADCH:ADCL`
  - 16-bits, 6-bits are unused
  - Option to adjust left or right
- 8 Analog input channels,
  - 7 differential input
  - with optional gain of 10x & 200x
  - However, only one conversion at a time
- V_{ref} options,

ATmega16/32 ADC Features

- 10-bit ADC
- 2 output registers,
  - `ADCH:ADCL`
  - 16-bits, 6-bits are unused
  - Option to adjust left or right
- 8 Analog input channels,
  - 7 differential input
  - with optional gain of 10x & 200x
  - However, only one conversion at a time
- V_{ref} options,
  - Analog V_{cc}, $5V$

ATmega16/32 ADC Features

- 10-bit ADC
- 2 output registers,
  - `ADCH:ADCL`
  - 16-bits, 6-bits are unused
  - Option to adjust left or right
- 8 Analog input channels,
  - 7 differential input
  - with optional gain of 10x & 200x
  - However, only one conversion at a time
- V_{ref} options,
  - Analog V_{cc}, $5V$
  - internal $2.56V$

ATmega16/32 ADC Features

- 10-bit ADC
- 2 output registers,
  - `ADCH:ADCL`
  - 16-bits, 6-bits are unused
  - Option to adjust left or right
- 8 Analog input channels,
  - 7 differential input
  - with optional gain of 10x & 200x
  - However, only one conversion at a time
- V_{ref} options,
  - Analog V_{cc}, $5V$
  - internal $2.56V$
  - external *AREF* pin

ATmega16/32 ADC Features

- 10-bit ADC
- 2 output registers,
  - `ADCH:ADCL`
  - 16-bits, 6-bits are unused
  - Option to adjust left or right
- 8 Analog input channels,
  - 7 differential input
  - with optional gain of 10x & 200x
  - However, only one conversion at a time
- V_{ref} options,
  - Analog V_{cc}, $5V$
  - internal $2.56V$
  - external *AREF* pin
- ADC clock rate != MCU CPU clock rate

ATmega16/32 ADC Features

- 10-bit ADC
- 2 output registers,
  - `ADCH:ADCL`
  - 16-bits, 6-bits are unused
  - Option to adjust left or right
- 8 Analog input channels,
  - 7 differential input
  - with optional gain of 10x & 200x
  - However, only one conversion at a time
- V_{ref} options,
  - Analog V_{cc}, $5V$
  - internal $2.56V$
  - external *AREF* pin
- ADC clock rate != MCU CPU clock rate
  - selection of pre-scaler

ATmega16/32 ADC Features

- 10-bit ADC
- 2 output registers,
  - `ADCH:ADCL`
  - 16-bits, 6-bits are unused
  - Option to adjust left or right
- 8 Analog input channels,
  - 7 differential input
  - with optional gain of 10x & 200x
  - However, only one conversion at a time
- V_{ref} options,
  - Analog V_{cc}, $5V$
  - internal $2.56V$
  - external *AREF* pin
- ADC clock rate != MCU CPU clock rate
  - selection of pre-scaler
  - AD Conversion takes at-least 13 ADC clock cycles

ADC Programming

Major Relevant registers

ADC Programming

Major Relevant registers

- `ADCH:ADCL`

ADC Programming

Major Relevant registers

- `ADCH:ADCL`
- `ADCSRA`

ADC Programming

Major Relevant registers

- `ADCH:ADCL`
- `ADCSRA`
- `ADMUX`

ADC Programming

Major Relevant registers

- `ADCH:ADCL`
- `ADCSRA`
- `ADMUX`
- `SFIOR`

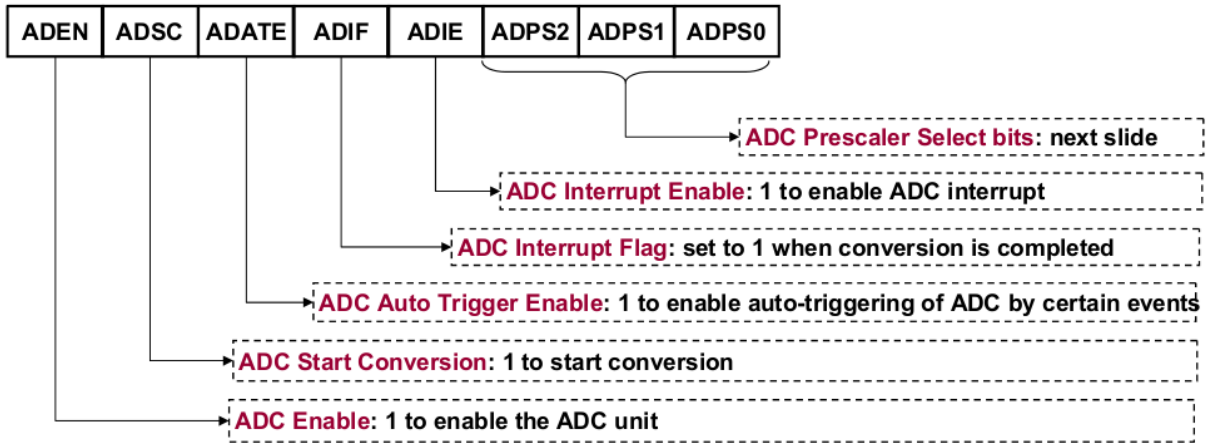
ADC Programming

Major Relevant registers

- `ADCH:ADCL`
- `ADCSRA`
- `ADMUX`
- `SFIOR`

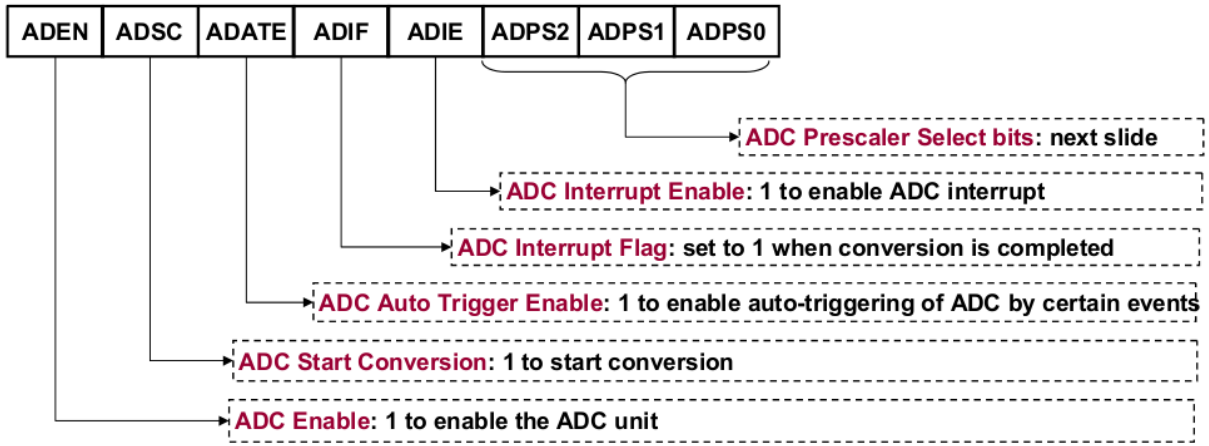
ADC Programming

ADCSRA Register



ADC Programming

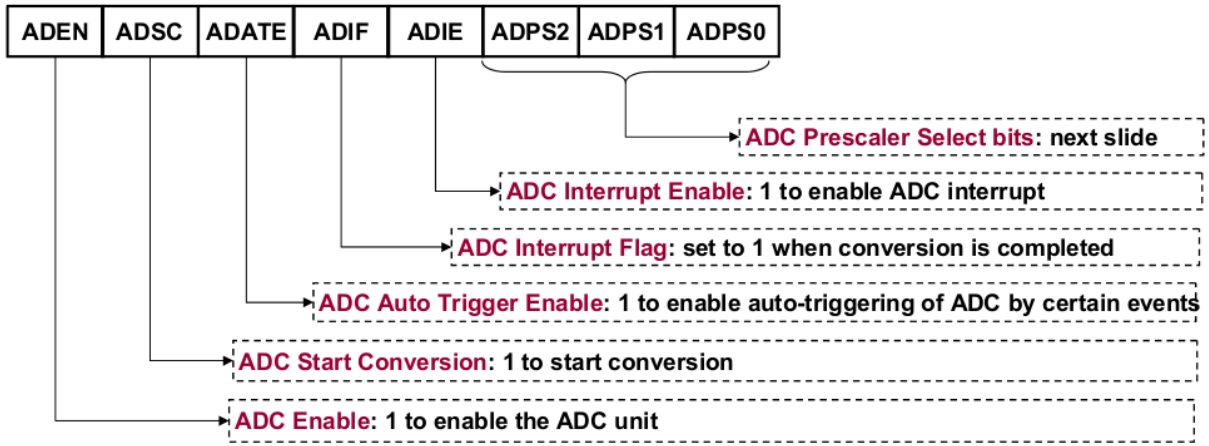
ADCSRA Register



ADPS2	ADPS1	ADPS0	ADC Clock
0	0	0	Reserved
0	0	1	CK/2
0	1	0	CK/4
0	1	1	CK/8
1	0	0	CK/16
1	0	1	CK/32
1	1	0	CK/64
1	1	1	CK/128

ADC Programming

ADCSRA Register

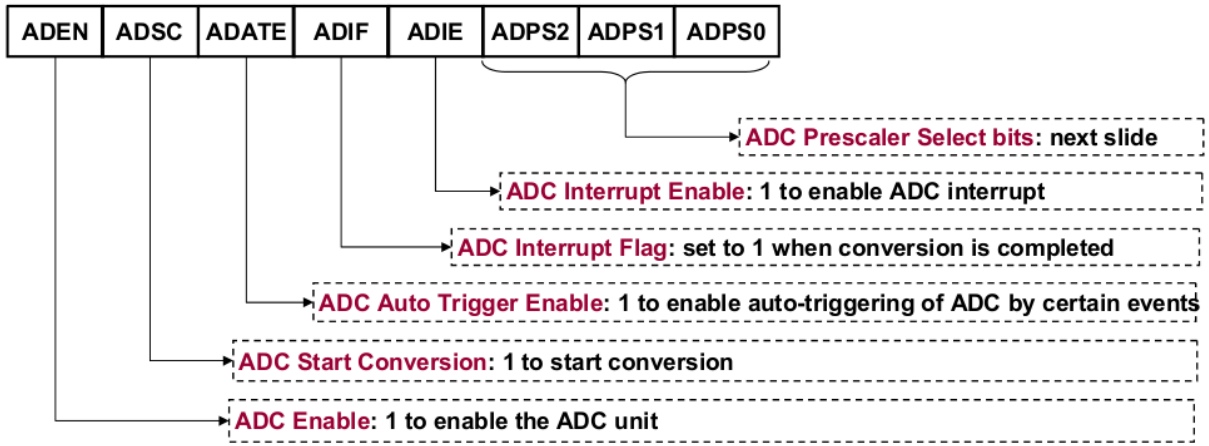


ADPS2	ADPS1	ADPS0	ADC Clock
0	0	0	Reserved
0	0	1	CK/2
0	1	0	CK/4
0	1	1	CK/8
1	0	0	CK/16
1	0	1	CK/32
1	1	0	CK/64
1	1	1	CK/128

- Initialization,

ADC Programming

ADCSRA Register

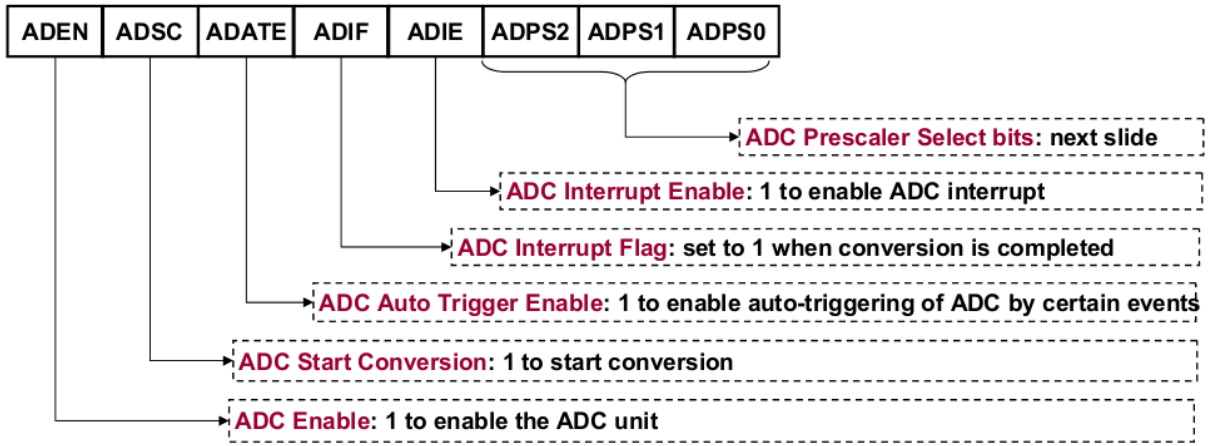


ADPS2	ADPS1	ADPS0	ADC Clock
0	0	0	Reserved
0	0	1	CK/2
0	1	0	CK/4
0	1	1	CK/8
1	0	0	CK/16
1	0	1	CK/32
1	1	0	CK/64
1	1	1	CK/128

- Initialization,
  - Polling, `ADCSRA = 0b10000001;`

ADC Programming

ADCSRA Register

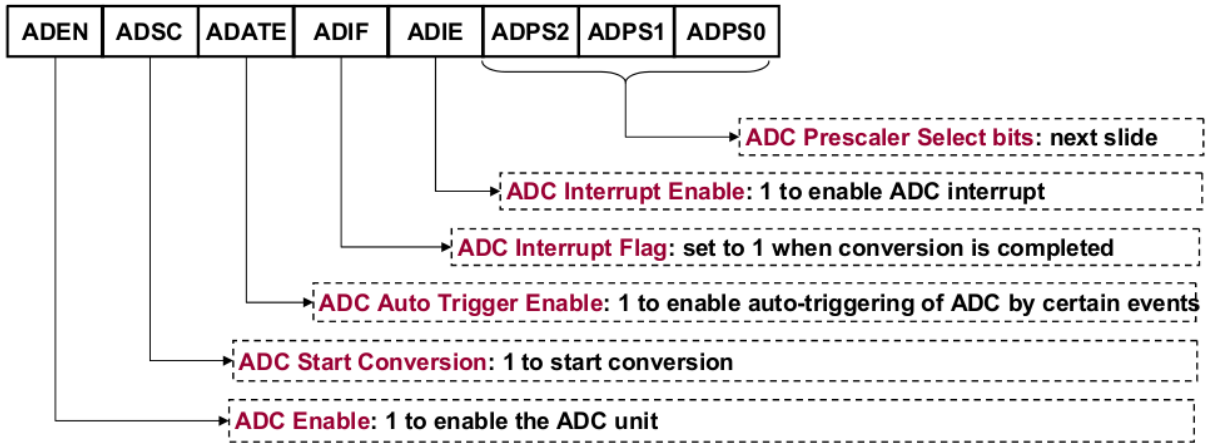


ADPS2	ADPS1	ADPS0	ADC Clock
0	0	0	Reserved
0	0	1	CK/2
0	1	0	CK/4
0	1	1	CK/8
1	0	0	CK/16
1	0	1	CK/32
1	1	0	CK/64
1	1	1	CK/128

- Initialization,
  - Polling, `ADCSRA = 0b10000001;`
  - Interrupt, `ADCSRA = 0b10001001;`

ADC Programming

ADCSRA Register

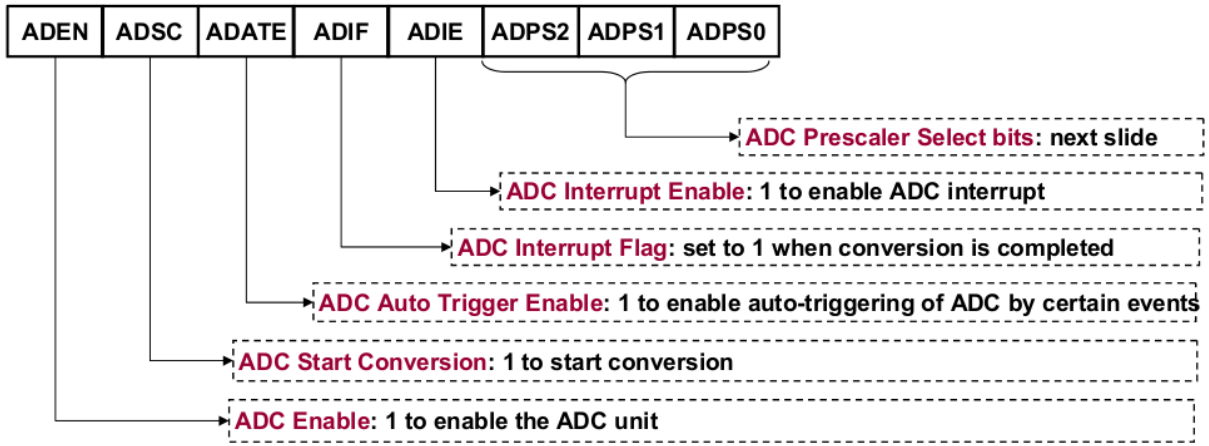


ADPS2	ADPS1	ADPS0	ADC Clock
0	0	0	Reserved
0	0	1	CK/2
0	1	0	CK/4
0	1	1	CK/8
1	0	0	CK/16
1	0	1	CK/32
1	1	0	CK/64
1	1	1	CK/128

- Initialization,
  - Polling, `ADCSRA = 0b10000001;`
  - Interrupt, `ADCSRA = 0b10001001;`
- Conversion Start,

ADC Programming

ADCSRA Register

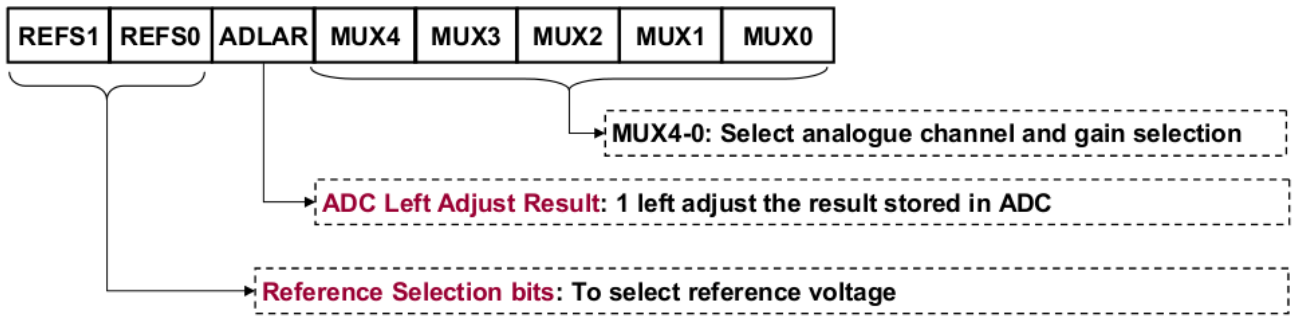


ADPS2	ADPS1	ADPS0	ADC Clock
0	0	0	Reserved
0	0	1	CK/2
0	1	0	CK/4
0	1	1	CK/8
1	0	0	CK/16
1	0	1	CK/32
1	1	0	CK/64
1	1	1	CK/128

- Initialization,
  - Polling, `ADCSRA = 0b10000001;`
  - Interrupt, `ADCSRA = 0b10001001;`
- Conversion Start,
  - `ADCSRA = ADCSRA | 0b01000000;`

ADC Programming

ADMUX Register



ADC Programming

ADMUX Register

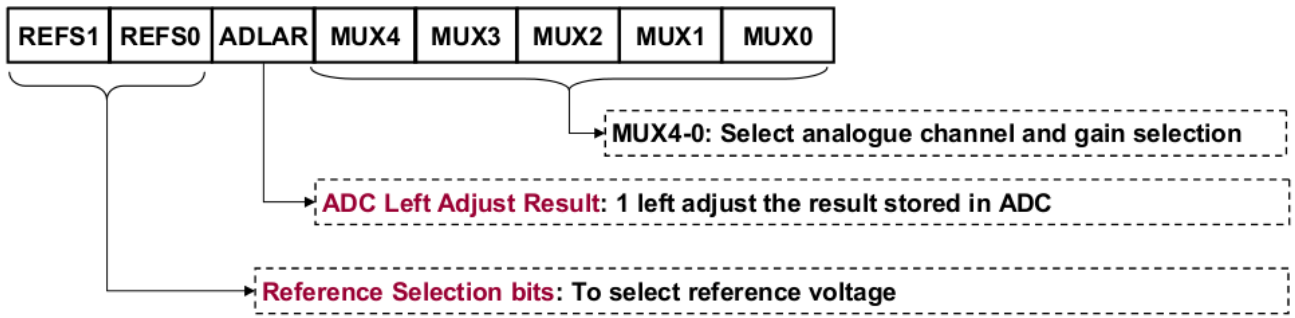


Table 13-4: V_{ref} Source Selection Table for AVR

REFS1	REFS0	V_{ref}	
0	0	AREF pin	Set externally
0	1	AVCC pin	Same as VCC
1	0	Reserved	----
1	1	Internal 2.56 V	Fixed regardless of VCC value

ADC Programming

ADMUX Register

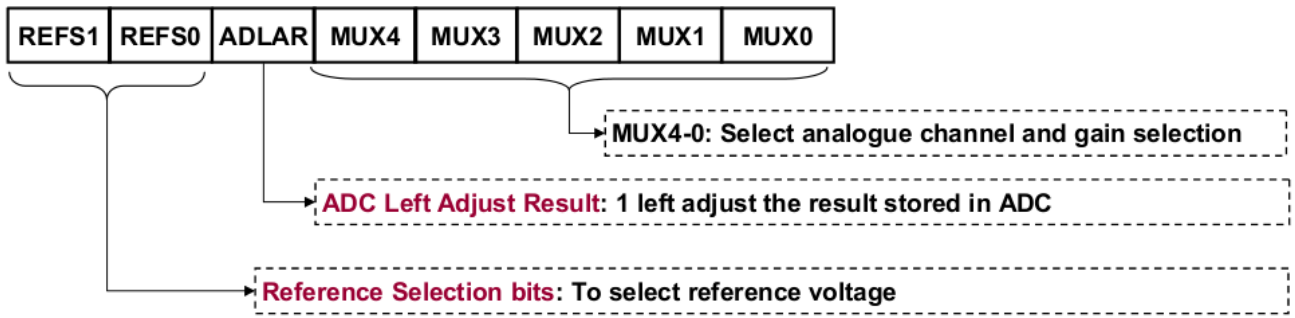
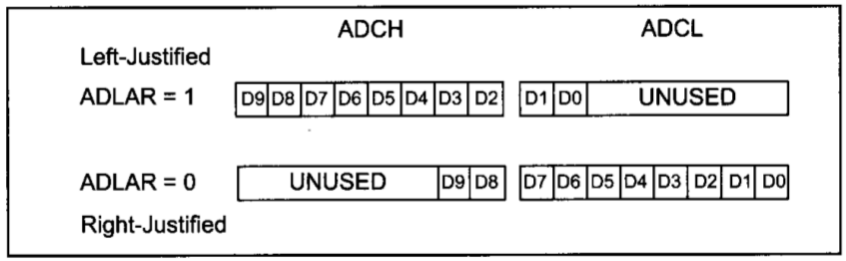


Table 13-4: V_{ref} Source Selection Table for AVR

REFS1	REFS0	V_{ref}	
0	0	AREF pin	Set externally
0	1	AVCC pin	Same as VCC
1	0	Reserved	----
1	1	Internal 2.56 V	Fixed regardless of VCC value



Sample ADC Program

Polling

```
 2
 3 #include <avr/io.h>
 4
 5 int main(void)
 6 {
 7     ADCSRA = 0b10000001;
 8     ADMUX = 0b11100000;
 9
10     DDRA = DDRB = 0b11111111;
11
12     while(1)
13     {
14         ADCSRA = ADCSRA | 0b01000000;
15         while((ADCSRA & 0b00010000) == 0){}
16         PORTA = ADCH;
17         PORTB = ADCL;
18     }
19
20     return 0;
21 }
```

Sample ADC Program

Interrupt

```
 2
 3 #include <avr/io.h>
 4 #include <avr/interrupt.h>
 5
 6 ISR(ADC_vect)
 7 {
 8     PORTA = ADCH;
 9     PORTB = ADCL;
10     ADCSRA = ADCSRA | 0b01000000;
11 }
12
13 int main(void)
14 {
15     ADCSRA = 0b10001001;
16     ADMUX = 0b11100000;
17
18     DDRA = DDRB = 0b11111111;
19     sei();
20     ADCSRA = ADCSRA | 0b01000000;
21
22     while(1)
23     {
24     }
25
26     return 0;
27 }
```

Reference

- The avr microcontroller & embedded system, *Chapter 13*
  - Muhammad Ali Mazidi
  - Sarmad Naimi
  - Sepehr Naimi