Chapter 4 Input/Output (I/O)

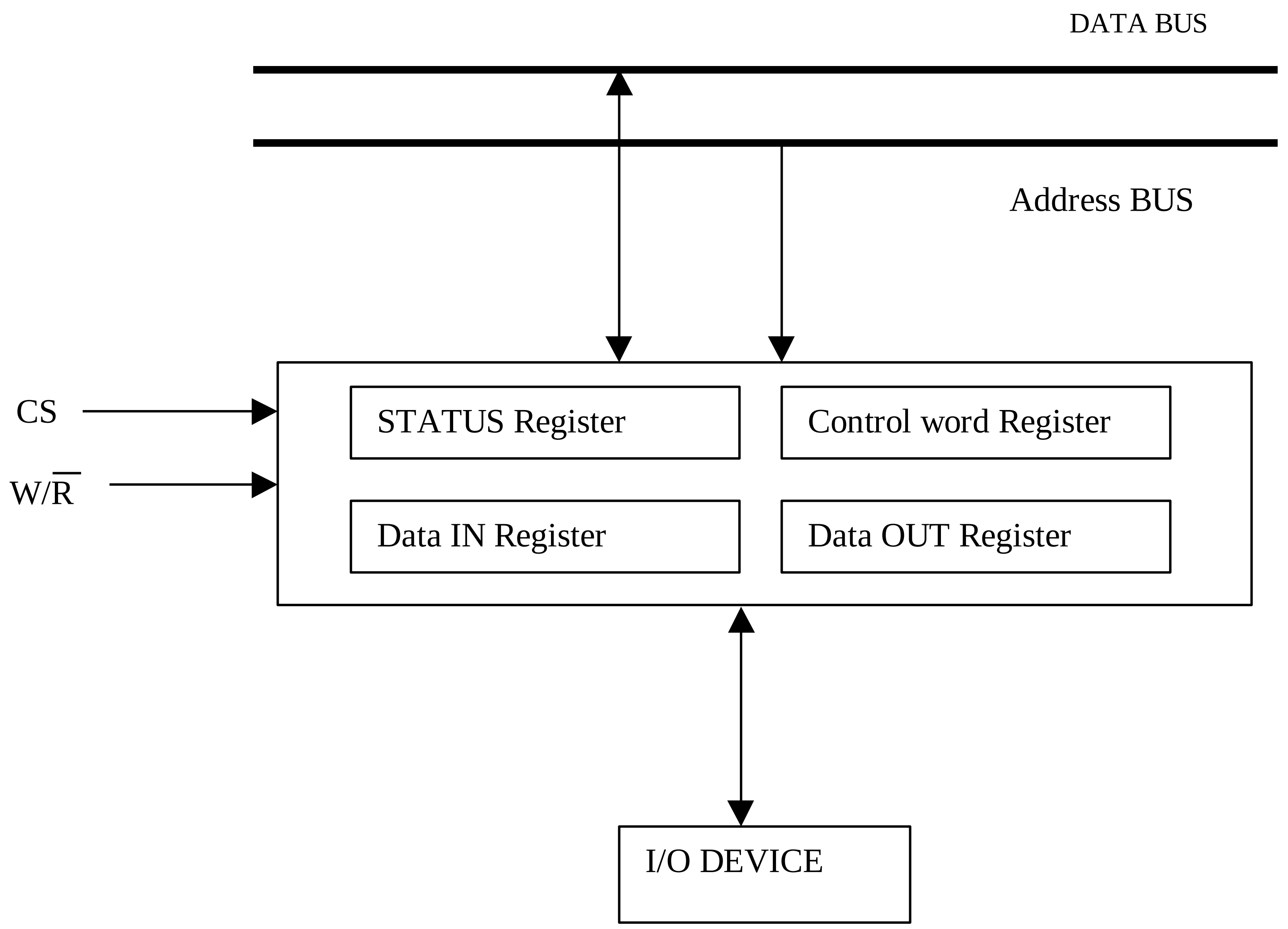
* peripheral
* programmed I/O
* interrupt
* timer

4.1 peripheral

- general concepts of peripheral
- digital-to-analog conversion (DAC)
- analog-to-digital conversion (ADC)

4.1.1 general concepts of peripheral

- there is an interface circuit between system bus and peripheral
- CPU reads or writes data to the device through the data IN/OUT registers
- status register reflects the status of the device
- CPU writes control words to the control word register to control the operation modes of the device
- each register occupies an address

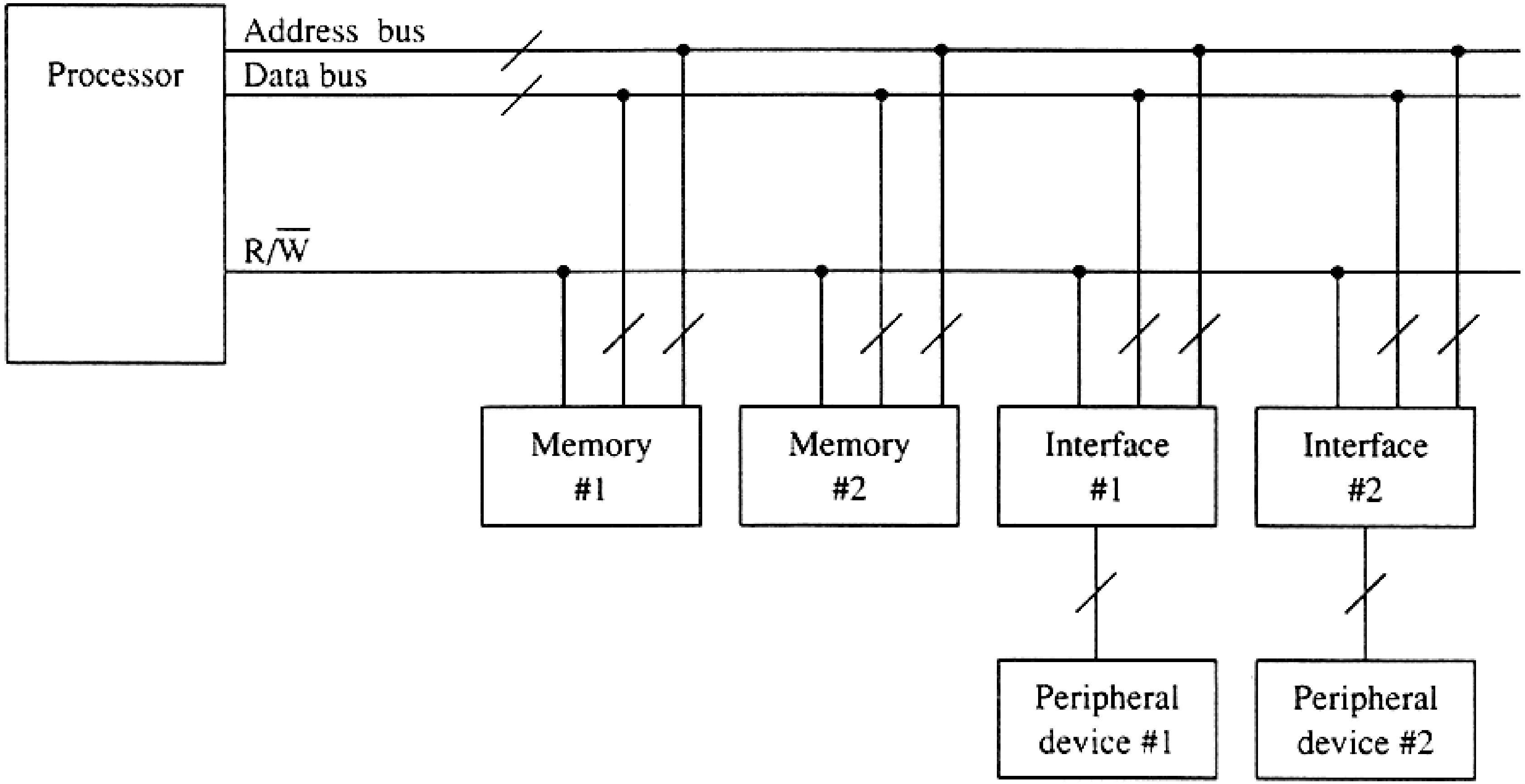

DATA BUS
Address BUS
CS
$W/\overline{R}$
STATUS Register
Control word Register
Data IN Register
Data OUT Register
I/O DEVICE

- for PIC18, TRISA, TRISB, … are used to control the I/O ports for input or output

- input/output the data from/to the device by accessing PORTA, PORTB, …

Memory Mapped I/O

I/O devices are addressed and selected by decoders as if they were memory devices.



Advantages :

- addressed and selected like a memory device
- all the memory reference operations and addressing modes can be used for I/O devices.
- e.g.

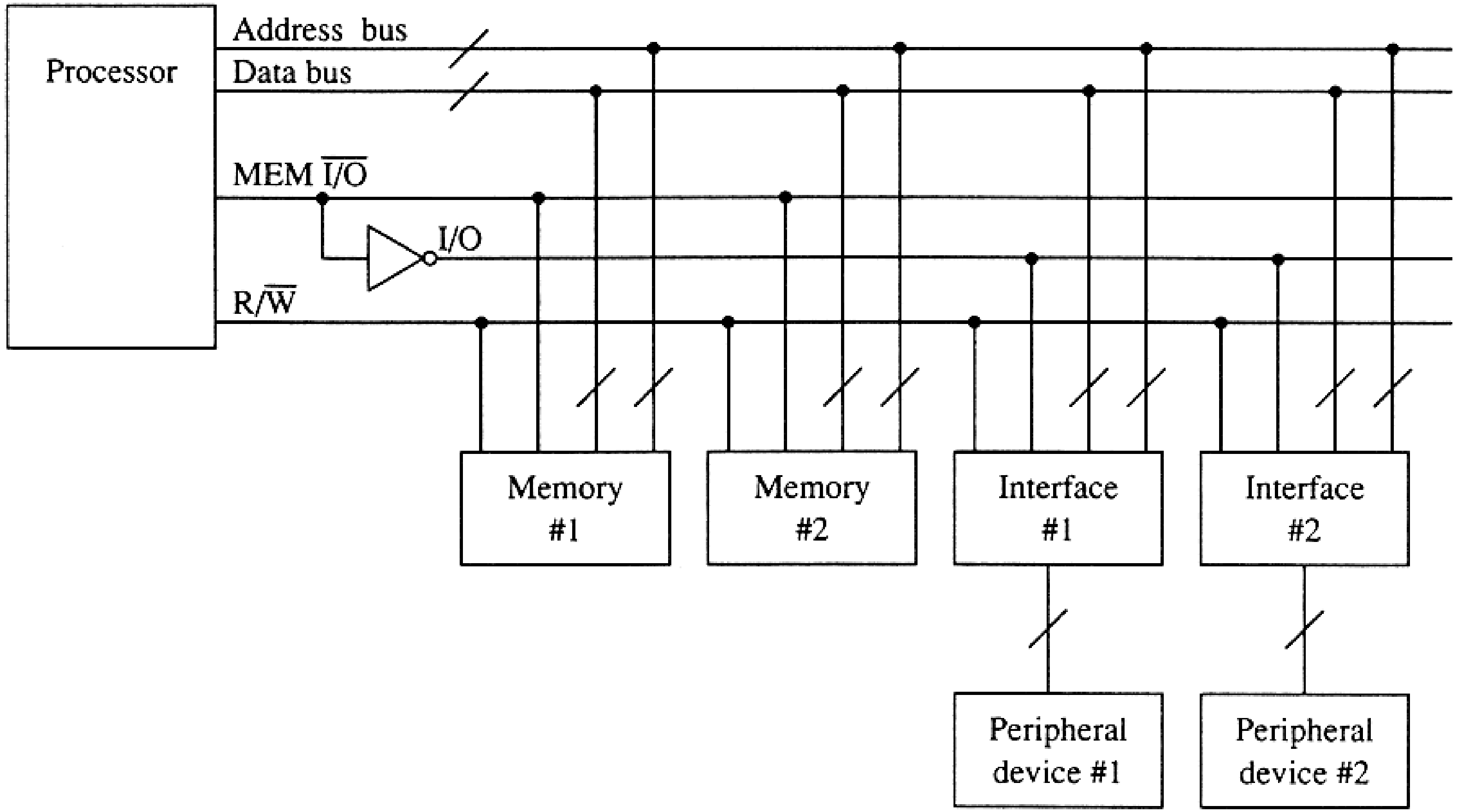
```
MOVWF TRISC
MOVF  PORTB, W
CLRF  TRISB
```

Disadvantages :

- some system memory addresses are used up for I/O and are therefore not available for memory
- the total space for memory is reduced

Direct I/O

Use separate address space with separate instructions and separate control signals for I/O.



Advantage:

- not using the system memory address space

Disadvantage:

- only special IN and OUT instructions can be used for I/O operations

e.g. In 8086

```
IN  AL, iox       ; iox address, range 00-FFh
OUT iox, AL       ; 8-bit I/O address
```

N.B. Some microprocessors, such as 8051, support only memory mapped I/O and some support both. The Intel 8086 supports both.

4.1.2 digital-to-analog conversion (DAC)

- analog signals are continuous, with infinite values in a given range

- examples: a voltmeter with a needle, music from speakers

- digital signals have discrete values - on/off, 0 or 1

- examples: digital voltmeter, digital watch

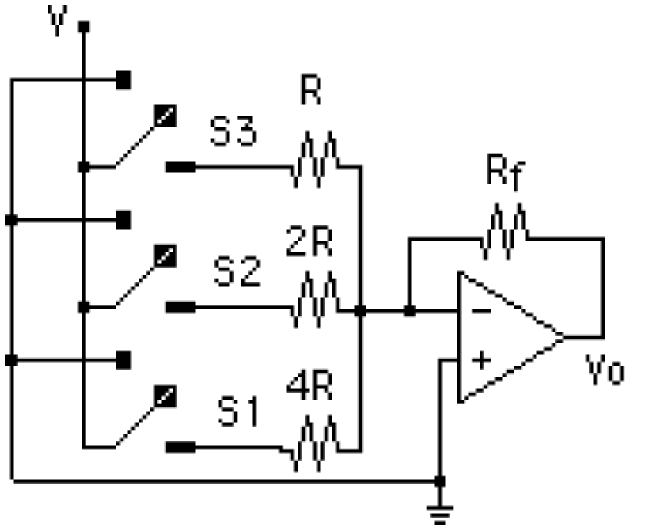
Data Conversion

- Limitations of analog signals
  - Analog signals pick up noise as they are being amplified.
  - Analog signals are difficult to store.
- Advantages of digital systems (signals)
  - Noise can be reduced by converting analog signals to 0s and 1s.
  - Binary signals can be easily stored in memory.
- The major limitation of a digital system is how accurately it represents the analog signals after conversion.

- embedded system often needs to convert signals from analog to digital, and back to analog
- the system includes:
  - a transducer that converts non-electrical signals into electrical signals (*velocity sensor*)
  - an A/D converter that converts analog signals into digital signals
  - microprocessor/microcontroller to process digital data
  - a D/A converter that converts digital signals into equivalent analog signals
  - a transducer that converts electrical signals into non-electrical signals (*brake*)

DAC

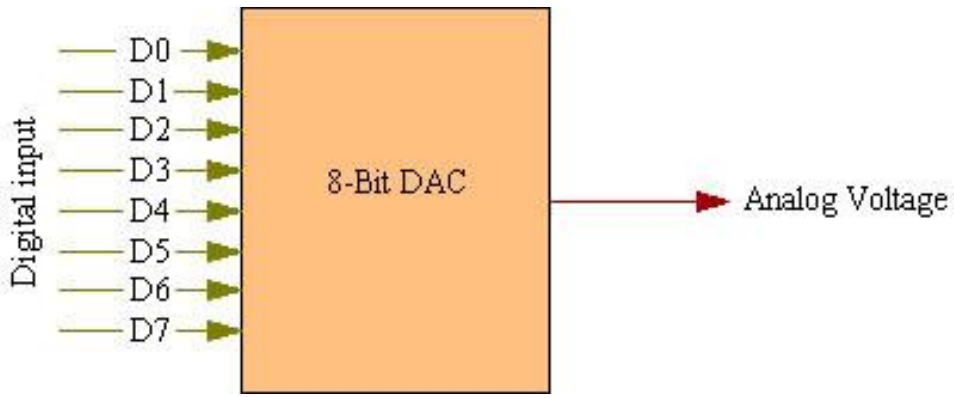
- use current source with digital switches



$$V0 = -R_f\left(S3\frac{V}{R} + S2\frac{V}{2R} + S1\frac{V}{4R}\right)$$

$$= \frac{-R_f}{4R}V\left(4S3 + 2S2 + S1\right)$$

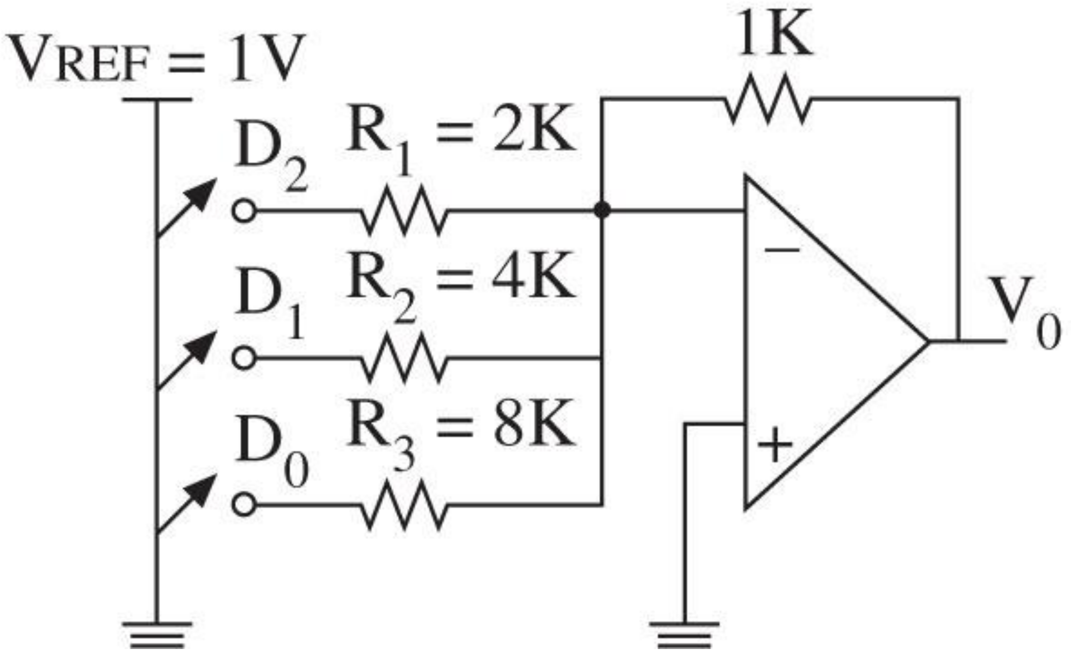
	S3	S2	S1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1



- convert discrete signals into discrete analog values that represent the magnitude of the input signal compared to a standard or reference voltage

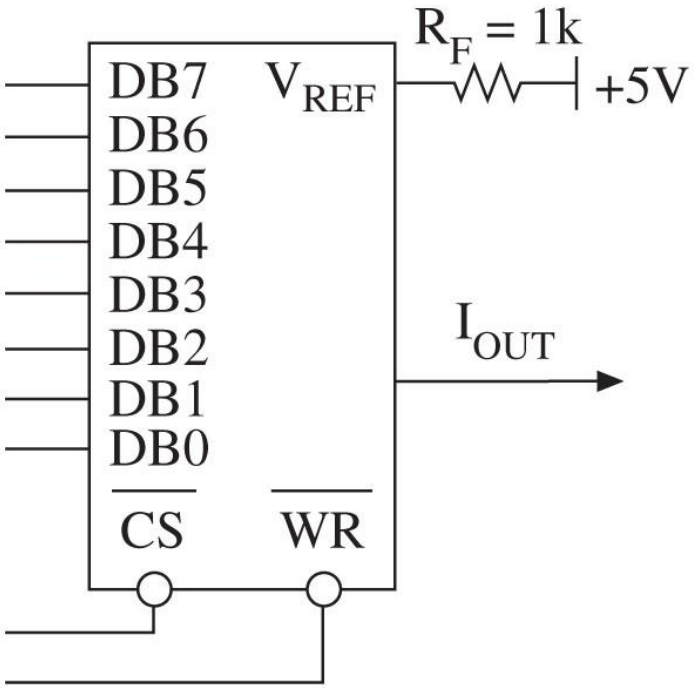
- the output of the DAC is discrete analog steps

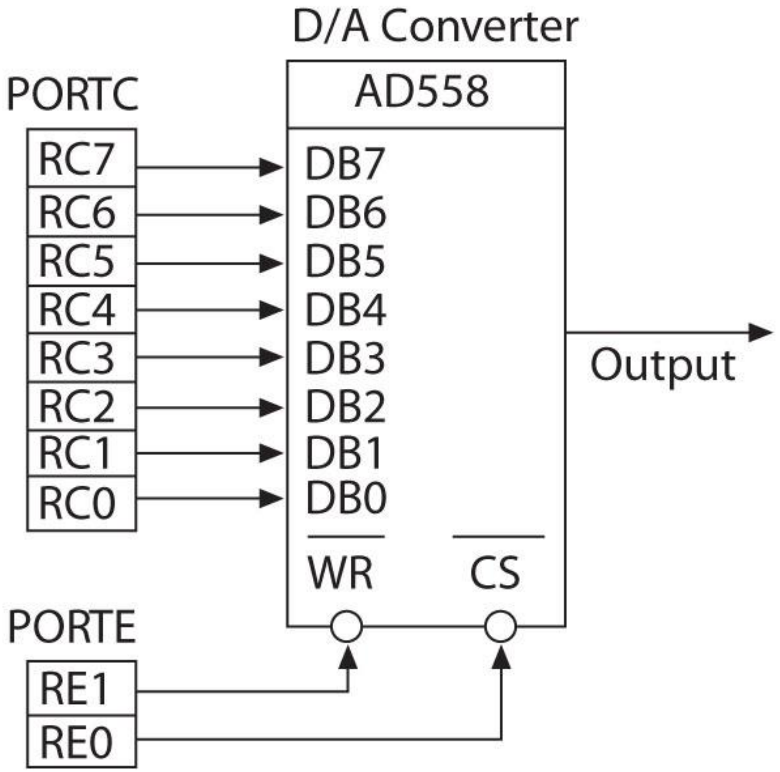
- by increasing the resolution (number of bits), the step size is reduced, and the output approximates a continuous analog signal



$$I_o = I_T = I_1 + I_2 + I_3 = \frac{V_{REF}}{R_1} + \frac{V_{REF}}{R_2} + \frac{V_{REF}}{R_3} = \frac{V_{REF}}{1k}\left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8}\right) = 0.875\,mA$$

$$V_O = -R_f I_T = -(1k)\times(0.875\ mA) = -0.875\ V = \left| {}^{7}\!/_{8}\ V \right|$$





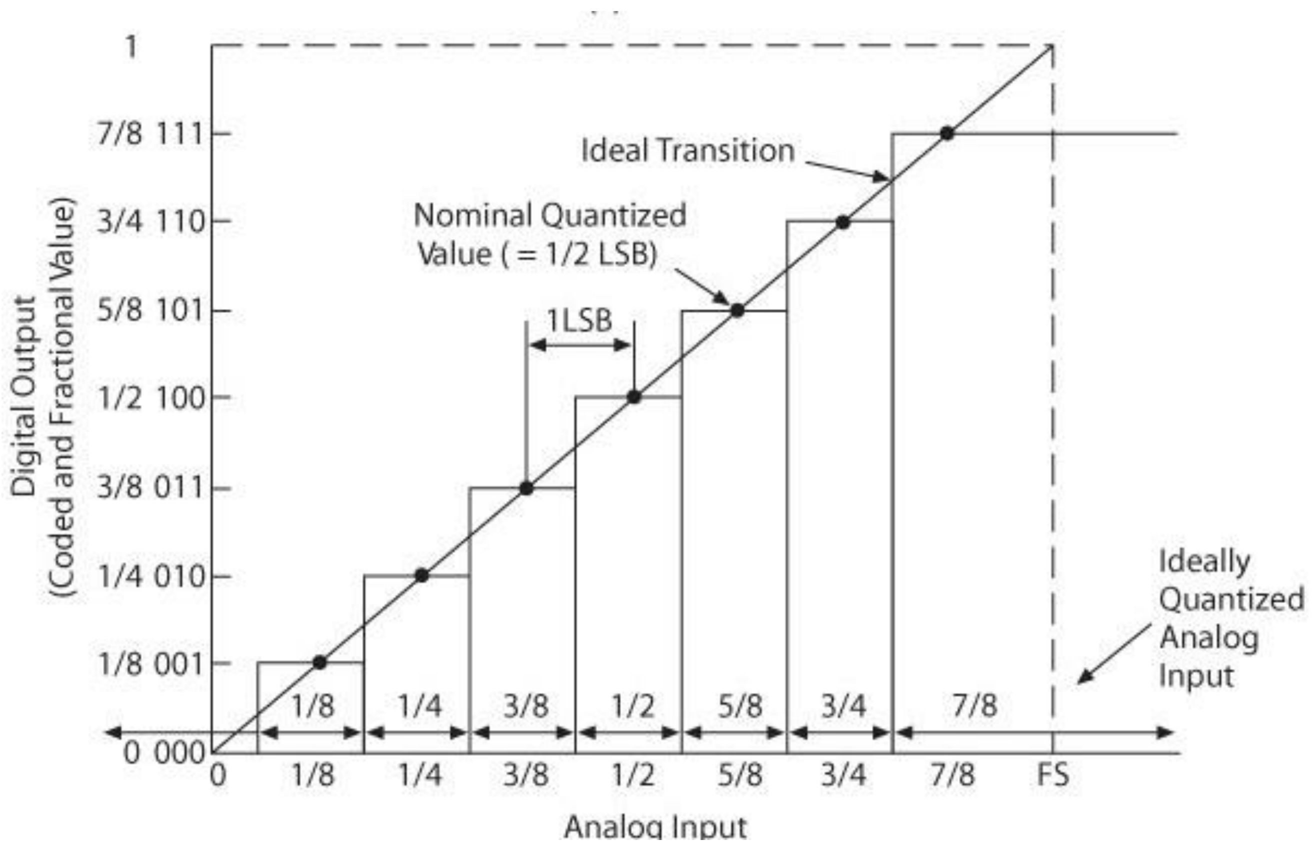
$$I_O = \frac{V_{REF}}{R_{REF}}\left(\frac{DB7}{2} + \frac{DB6}{4} + \frac{DB5}{8} + \ldots\ldots + \frac{DB0}{2^8}\right)$$

$$I_O = 5\ \text{mA}\left(\frac{1}{2} + \frac{0}{4} + \frac{0}{8} + \frac{1}{16} + \frac{0}{32} + \frac{0}{64} + \frac{0}{128} + \frac{1}{256}\right) = 2.832\ \text{mA}$$

Write a program to generate a sine wave with angles between 0 and 360 degrees (with 30-degree increments). The full-scale output of the DAC is 10V. Output the sine wave to PORT B.

4.1.3 analog-to-digital conversion (ADC)

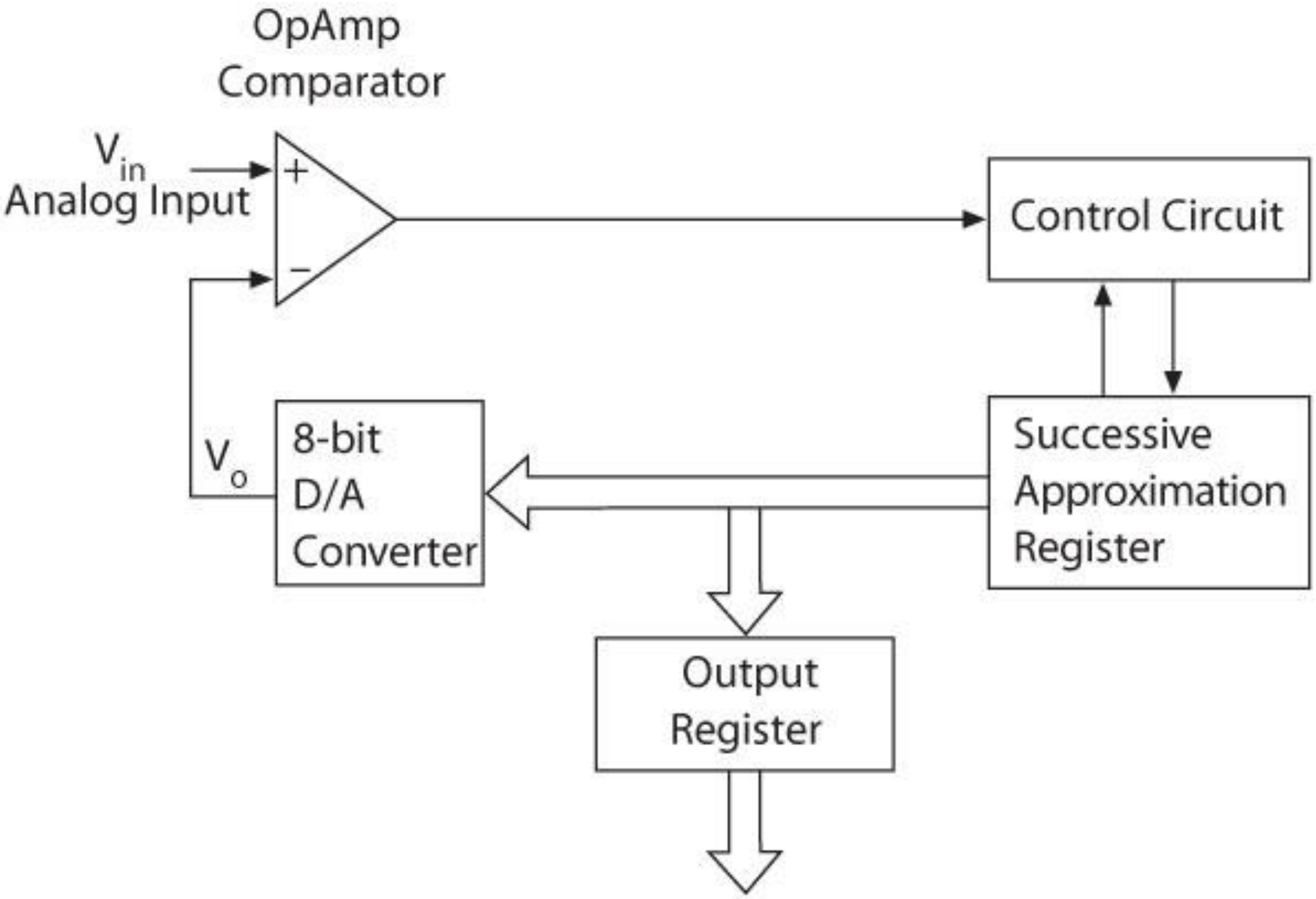
- convert a continuous varying signal, such as voltage or current, into discrete digital quantities that represent the magnitude of the signal compared to standard or reference voltage



Data Conversion

- Flash
  - Uses multiple comparators in parallel
  - High-speed, high-cost converter
- Integrator
  - Charges a capacitor for a given amount of time using the analog signal
  - Slow, but high accuracy and low noise
- Successive Approximation
  - Effective compromise among resolution, speed, and cost
- Counter
  - Similar to successive approximation circuit
  - Slower, with variable conversion times

Successive Approximation



Major Characteristics

- Resolution
  - n-bit digital output, n can be 8, 12, …
  - higher resolution, smaller step size (8-bit, V_{ref} = 5V, step size = 19.53 mV)
- Conversion time
  - time to convert the analog input to the digital output
  - depend on clock, conversion method, ADC chip technology
- V_{ref}
  - lower voltage, smaller step size (8-bit, V_{ref} = 4V, step size = 15.62 mV)
  - sometimes, differential reference voltage is used, $V_{ref} = V_{ref}(+) - V_{ref}(-)$
  - $V_{ref}(-)$ pin is connected to ground, $V_{ref}(+)$ is used as V_{ref}

- Digital output
  - $D_{out} = \frac{V_{in}}{step\ size}$, D_{out} = digital output (in decimal)
  - e.g. 8-bit, V_{ref} = 2.56V, V_{in} = 1.7V, D_{out} = 170 (decimal), 10101010 (binary)
- Parallel versus serial
  - parallel: more data pins, shorter time to get data
  - serial: one data pin (parallel-in-serial-out shift register inside ADC), longer time to get data
- Analog input
  - ADC can have multiple input channels
  - can monitor multiple sensors

- Conversion signals
  - start conversion signal (SC), starts converting V_{in} to D_{out}
  - end-of-conversion signal (EOC), notifies CPU the converted data is ready to be picked up

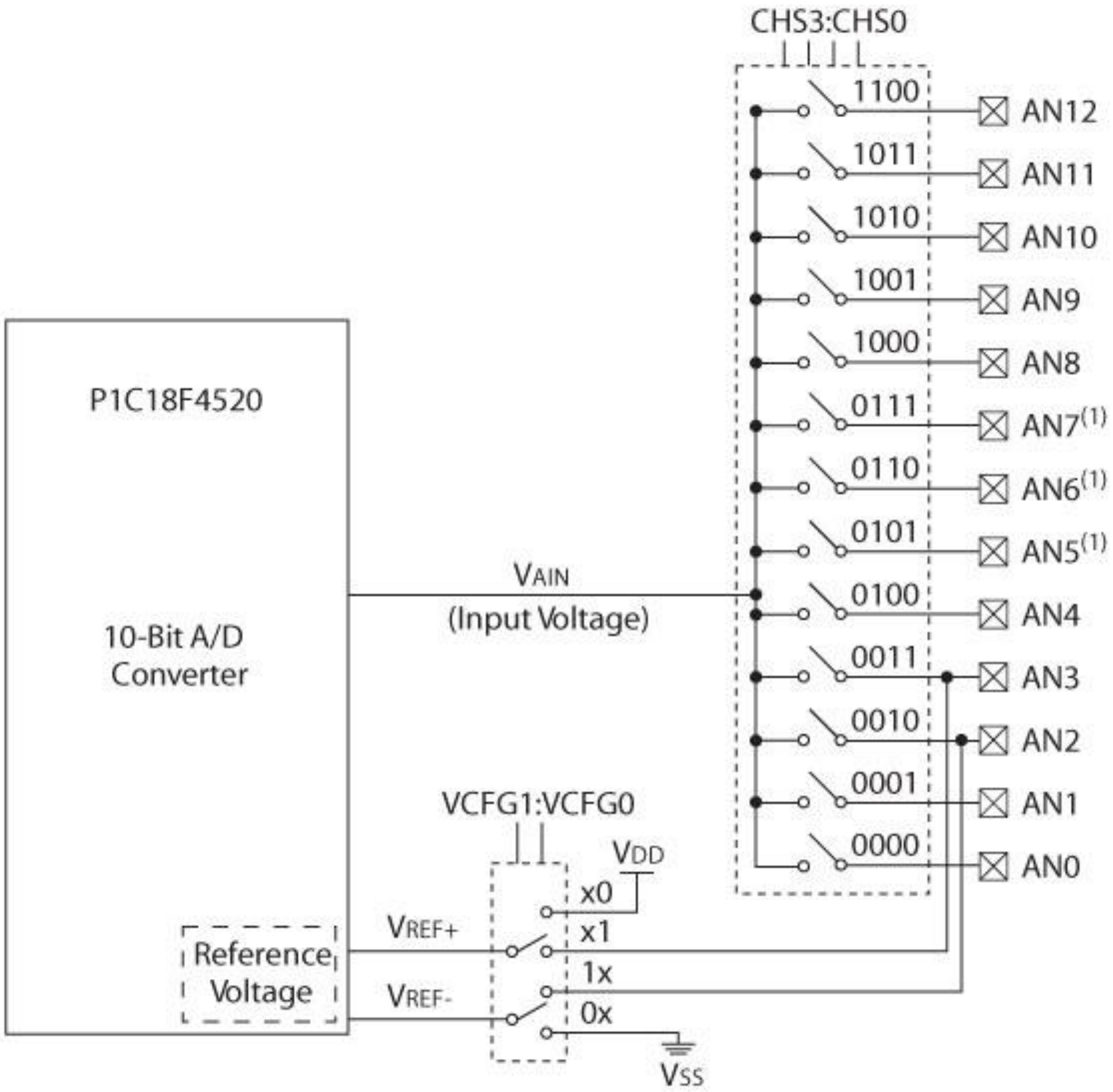
ADC procedure:

- select a channel
- activate SC signal
- monitor EOC signal
- read data from ADC

PIC18F A/D Converter Module

The PIC18F4520 microcontroller includes:

- 10-bit A/D converter
- 13 analog input channels AN0 - AN12
- 3 control and status registers: **ADCON0**, **ADCON1**, and **ADCON2**
- data registers: **ADRESH** (**A/D Res**ult **H**igh), **ADRESL** (**A/D Res**ult **L**ow), *6 bits unused*
- conversion time is dictated by crystal frequency F_{OSC}



CHS3:CHS0
1100
1011
1010
1001
1000
0111
0110
0101
0100
0011
0010
0001
0000
AN12
AN11
AN10
AN9
AN8
AN7(1)
AN6(1)
AN5(1)
AN4
AN3
AN2
AN1
AN0
P1C18F4520
10-Bit A/D
Converter
VAIN
(Input Voltage)
VCFG1:VCFG0
VDD
x0
x1
1x
0x
VREF+
VREF-
Reference
Voltage
VSS

A/D converter requires a low reference voltage (V_{REF-}) and a high reference voltage (V_{REF+}) to perform conversion.

1. An analog input of V_{REF-} is converted to digital code 0.
2. An analog input of V_{REF+} is converted to digital code $2^n - 1$.
3. An analog input of V_{in} is converted to digital code

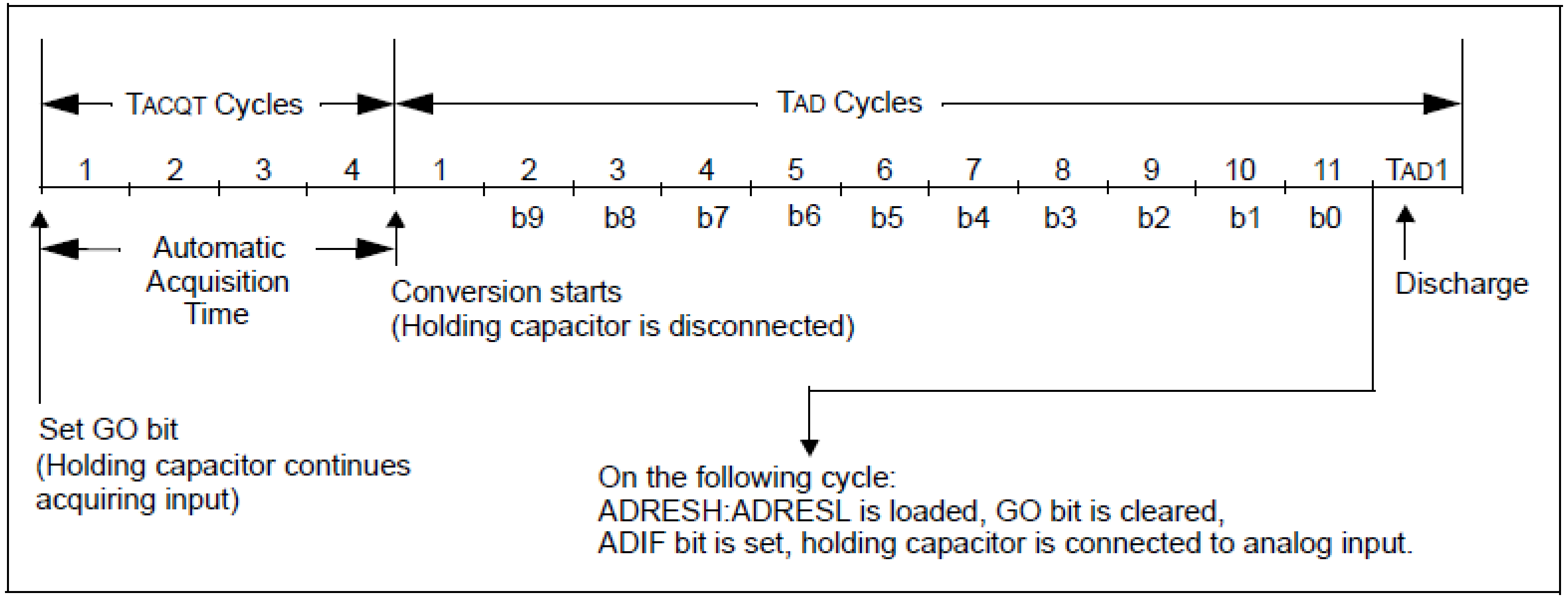
$$k = (2^n - 1) \times (V_{in} - V_{REF-}) \div (V_{REF+} - V_{REF-})$$

Given k, the measured voltage is given by

$$V_{in} = V_{REF-} + (V_{REF+} - V_{REF-}) \times k \div (2^n - 1)$$

Most systems use V_{DD} and 0V as V_{REF+} and V_{REF-}, respectively.

The output of a transducer should be scaled and shifted to the range of 0V ~ V_{DD} in order to achieve the best accuracy.



T_{AD} is the conversion time per bit

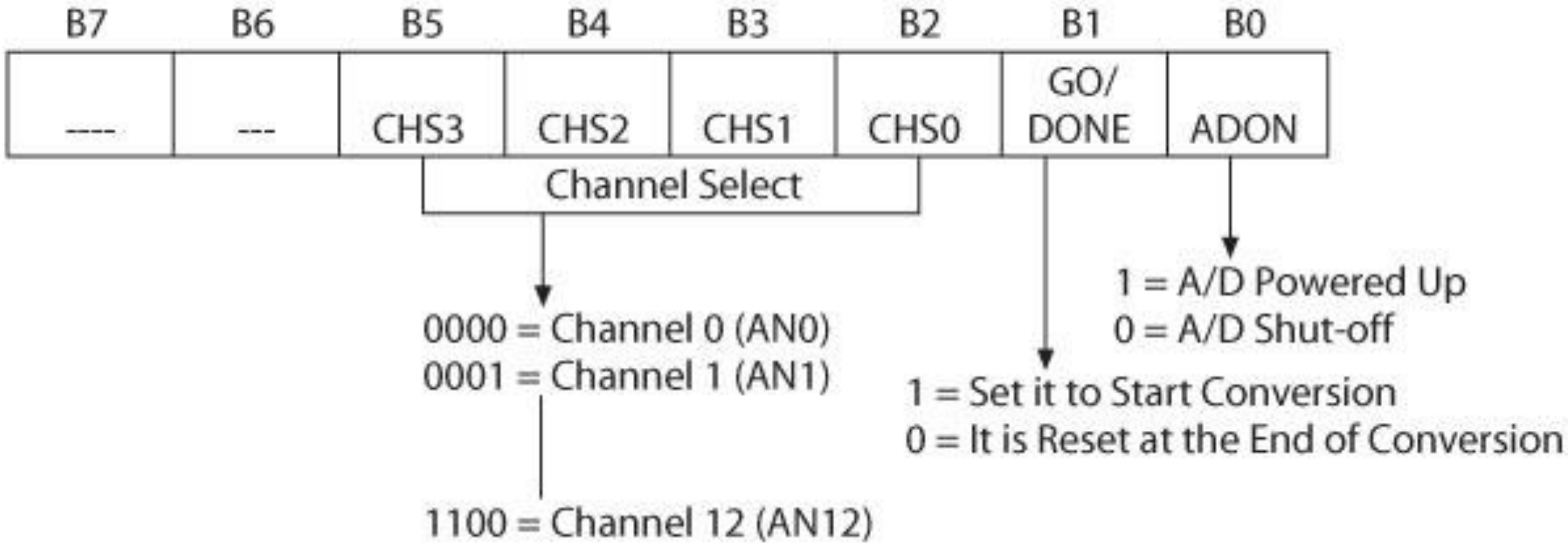
T_{ACQT} is the setup time for ADC

Both timing parameters can be programed

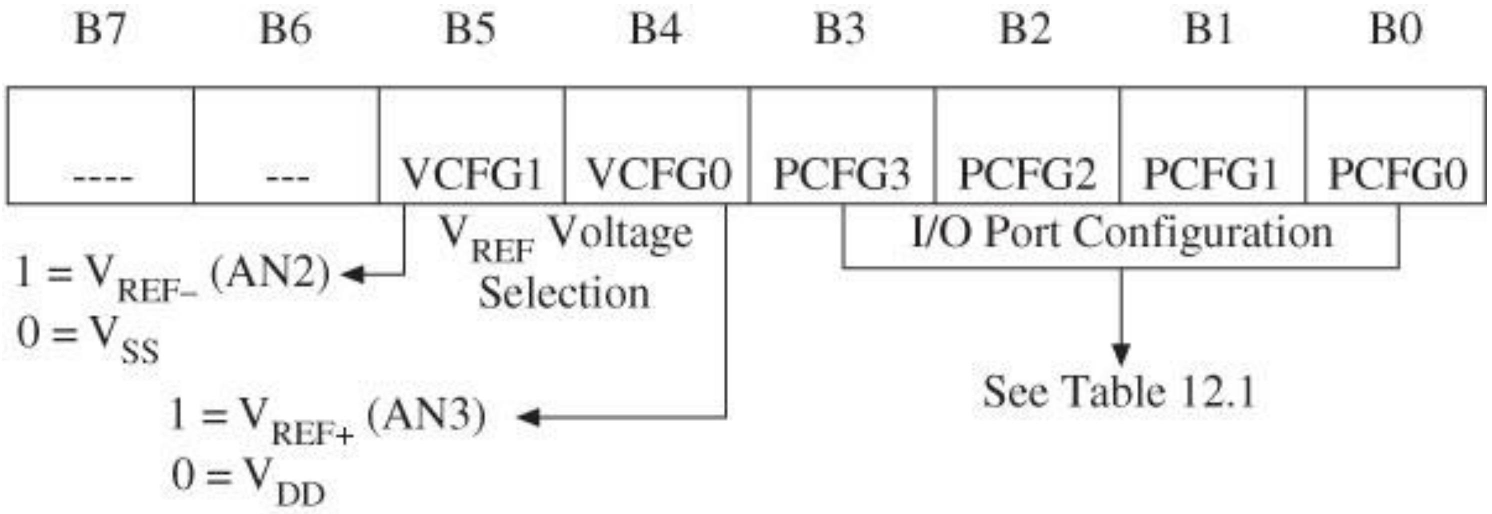
Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
130	T_{AD}	A/D Clock Period	PIC18**F**XXXX	0.7	25.0(1)	μs	T_{OSC} based, $V_{REF} \geq 3.0V$
			PIC18**LF**XXXX	1.4	25.0(1)	μs	$V_{DD} = 2.0V$; T_{OSC} based, V_{REF} full range
			PIC18**F**XXXX	TBD	1	μs	A/D RC mode
			PIC18**LF**XXXX	TBD	3	μs	$V_{DD} = 2.0V$; A/D RC mode
131	T_{CNV}	Conversion Time (not including acquisition time) **(Note 2)**		11	12	T_{AD}	
132	T_{ACQ}	Acquisition Time **(Note 3)**		1.4 TBD	— —	μs μs	-40°C to +85°C 0°C ≤ to ≤ +85°C
135	T_{SWC}	Switching Time from Convert → Sample		—	**(Note 4)**		
TBD	T_{DIS}	Discharge Time		0.2	—	μs	

A/D Control Register

- ADCON0 Register
  - Select analog input channel
  - Start a conversion
  - Indicate the end of the conversion



- ADCON1 Register
  - Set up the I/O pins either for analog signal or for digital signals
  - Select V_{REF} voltages

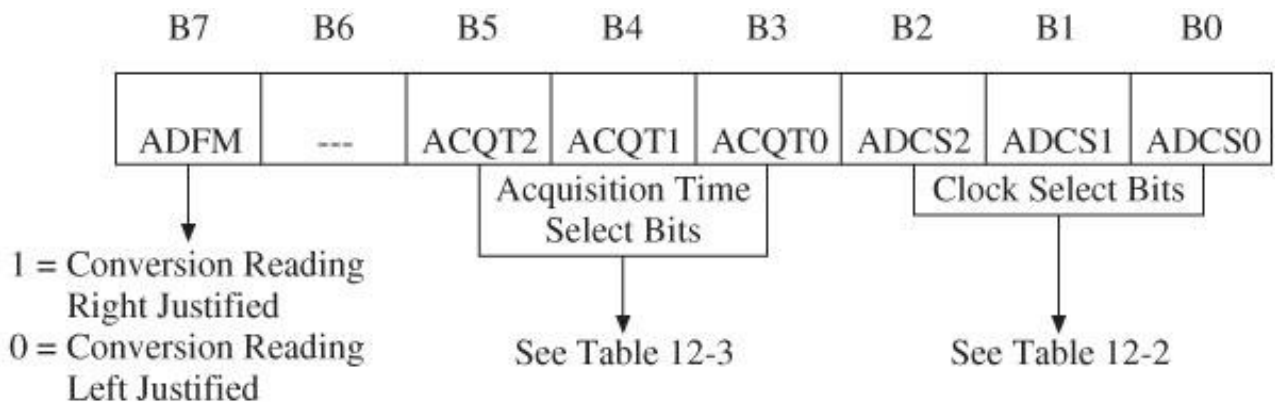


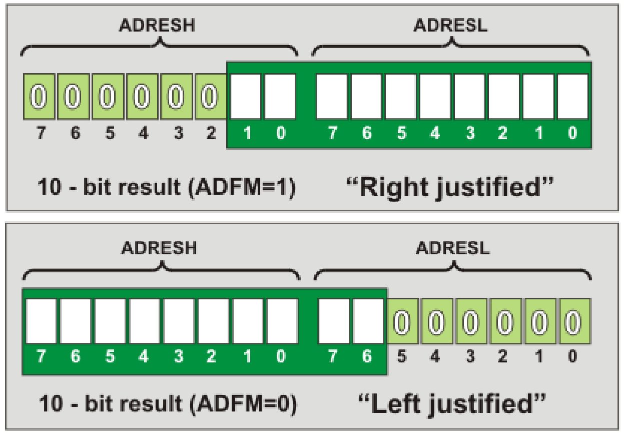
	7	6	5	4	3	2	1	0
	--	--	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
value after reset	0	0	0	0	0	0	0	0

VCFG1:VCFG0: Voltage reference configuration bits
(see Table 12.2)
PCFG3:PCFG0: A/D port configuration control bits

	AN15	AN14	AN13	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
0000	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A
0111	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

- ADCON2 Register
  - Select an acquisition time and clock frequency
  - Right or left justify output reading





ACQT2:ACQT0: A

000 = 0 TAD(1)
001 = 2 TAD
010 = 4 TAD
011 = 6 TAD
100 = 8 TAD
101 = 12 TAD
110 = 16 TAD
111 = 20 TAD

ADCS2:ADCS0: A/D conversion clock select bits

000 = FOSC/2
001 = FOSC/8
010 = FOSC/32
011 = FRC (clock derived from A/D RC oscillator)
100 = FOSC/4
101 = FOSC/16
110 = FOSC/64
111 = FRC (clock derived from A/D RC oscillator)

Selecting ADC conversion time

- The programming value of T_{AD} (ADCS2-ADCS0) must be greater than the minimum value of T_{AD}
- The programming value of T_{ACQ} (ACQT2-ACQT0) must be greater than the minimum of T_{ACQ}

Let

the minimum value of $T_{AD} = 1.6\ \mu s$
the minimum value of $T_{ACQ} = 13\ \mu s$
$f_{OSC} = 32$ MHz

What are (ADCS2-ADCS0) and (ACQS2-ADCS0) ?

- $f_{OSC} = 32$ MHz, the A/D clock source must be set to $f_{OSC}/64$ which makes $T_{AD} = 2\mu s$. (ADCS2-ADCS0) = 110
- For $T_{AD} = 2\mu s$, T_{ACQ} must be set to at least 8 T_{AD}.

(ACQT2-ACQT0) = 100

Assembly instruction sequence that achieve the desired setting:

```
movlw    0x01        ; select channel AN0 and enable A/D
movwf    ADCON0,A
movlw    0x0E        ; configure only channel AN0 as analog port
movwf    ADCON1,A    ; select VDD and VSS as reference voltages
movlw    0xA6        ; set A/D result right justified, set acquisition
movwf    ADCON2,A    ; time to 8 TAD, clock source FOSC/64
```

Procedure for performing A/D Conversion

- Configure the A/D module
    1. Configure analog pins, reference voltages
    2. Select A/D input channel
    3. Select A/D acquisition time (if available)
    4. Select A/D conversion clock
    5. Enable A/D module
- Configure A/D interrupt
    1. Clear ADIF flag
    2. Set ADIE bit (if desired)
    3. Set GIE bit (if desired)
- Wait for the desired acquisition time (if required)
- Start conversion by setting the GO/DONE bit
- Wait for A/D conversion to complete
- Read the A/D result registers; clear the ADIF flag

Example

Assume that the AN0 pin of a PIC18 running with a 32 MHz crystal oscillator is connected to a potentiometer. The voltage range of the potentiometer is from 0V to 5V. Write a program to measure the voltage applied to the AN0 pin, convert it, and retrieve the conversion result and place it in PRODH:PRODL.

Let the minimum value of $T_{AD} = 1.6\ \mu s$ and the minimum value of $T_{ACQ} = 13\ \mu s$

```
       org 0x00
       goto start
       org 0x08
       retfie
       org 0x18
       retfie
Start: movlw 0x01     ; select channel AN0 and enable A/D
       movwf ADCON0,A ; "
       movlw 0x0E     ; use VDD & VSS as reference voltages &
       movwf ADCON1,A ; configure channel AN0 as analog input
       movlw 0xA6     ; select FOSC/64 as conversion clock,
       movwf ADCON2,A ; 8 TAD for acquisition time,
                      ;right-justified
       bsf ADCON0,GO,A ; start A/D conversion
wait_con:
       btfsc ADCON0,DONE,A ; wait until conversion is done
       bra wait_con
       movff ADRESH,PRODH ; save conversion result
       movff ADRESL,PRODL ;
end
```

Scaling circuit

Used to amplify the transducer output from a range of $0V \sim V_Z$ to $0 \sim V_{DD}$.

- Usually V_Z is smaller than V_{DD}.

- Voltage gain: $A_V = V_{OUT} \div V_{IN} = (R_1 + R_2) \div R_1 = 1 + R_2/R_1$

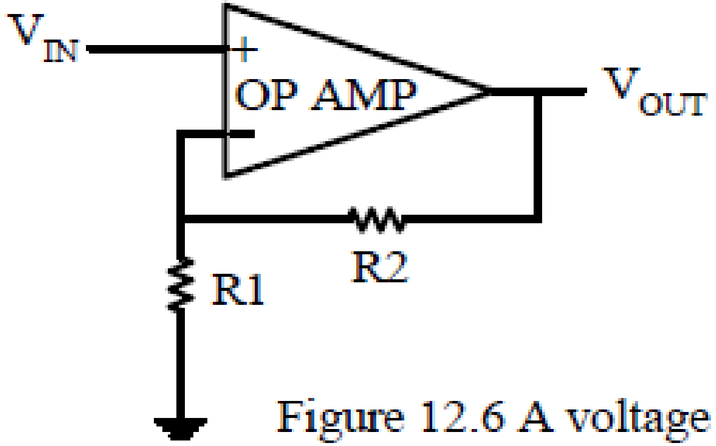


Figure 12.6 A voltage scaler

Suppose the transducer output voltage ranges from 0V to 200 mV.

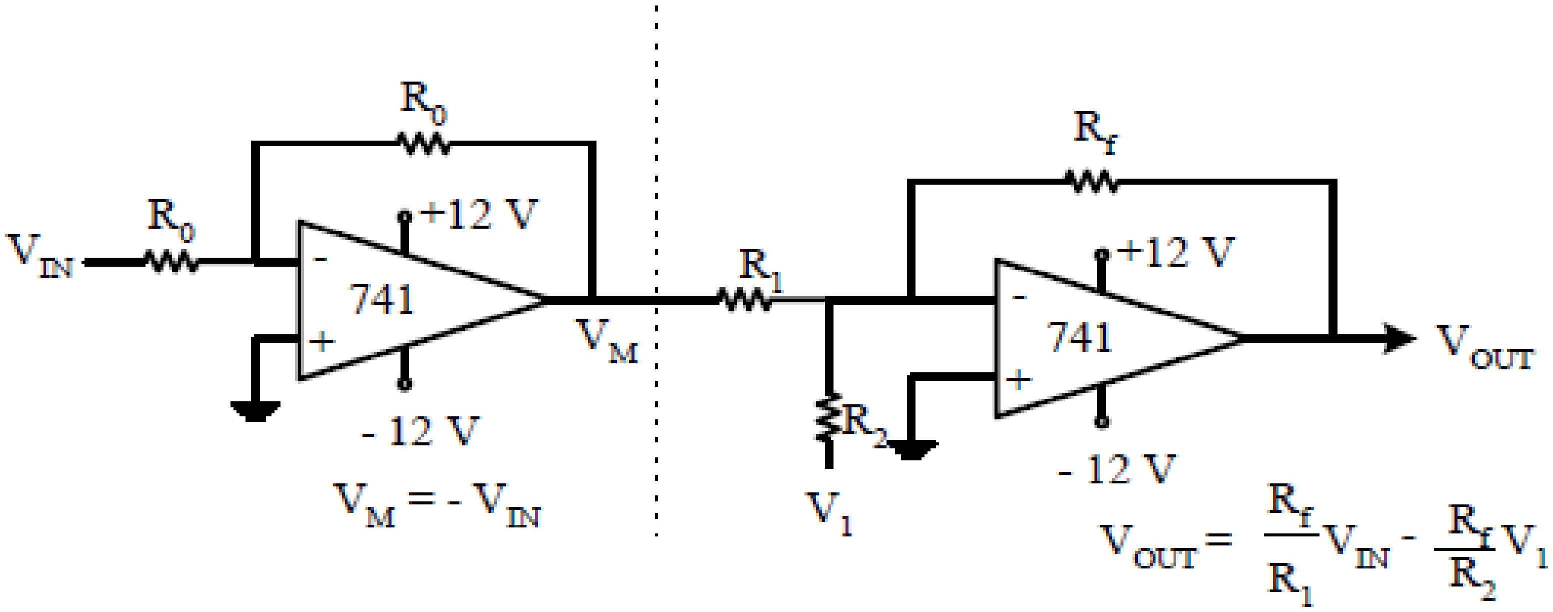
Choose the appropriate values for R1 and R2 to scale this range to 0 ~ 5V.

Solution:

- $R2/R1 = (V_{OUT} / V_{IN}) - 1 = 24$
- Choose 240 KΩ for R2 and 10 KΩ for R1.

Voltage translation circuit

Needed to shift and scale the transducer output in a range of $-V_X \sim V_Z$ to $0V \sim V_{DD}$



Choose appropriate resistor values and the adjusting voltage so that the circuit shown in the last page can shift the voltage from the range of –1.2V ~ 3.0V to the range of 0V ~ 5V.

Solution:

$$0 = -1.2 \times (Rf/R1) - (Rf/R2) \times V1$$
$$5 = 3.0 \times (Rf/R1) - (Rf/R2) \times V1$$

Choose $R_0 = R_1 = 10\ K\Omega$ and $V_1 = -5V$
$R_f = 12\ K\Omega$, $R_2 = 42\ K\Omega$

Summary

- general concepts of peripheral
- DAC
- ADC