EEC 417/517
Embedded Systems
Cleveland State University

Lab 2

The Analog to Digital (A/D) Conversion Module

The Timer0 Module

The `banksel` Directive

Dan Simon

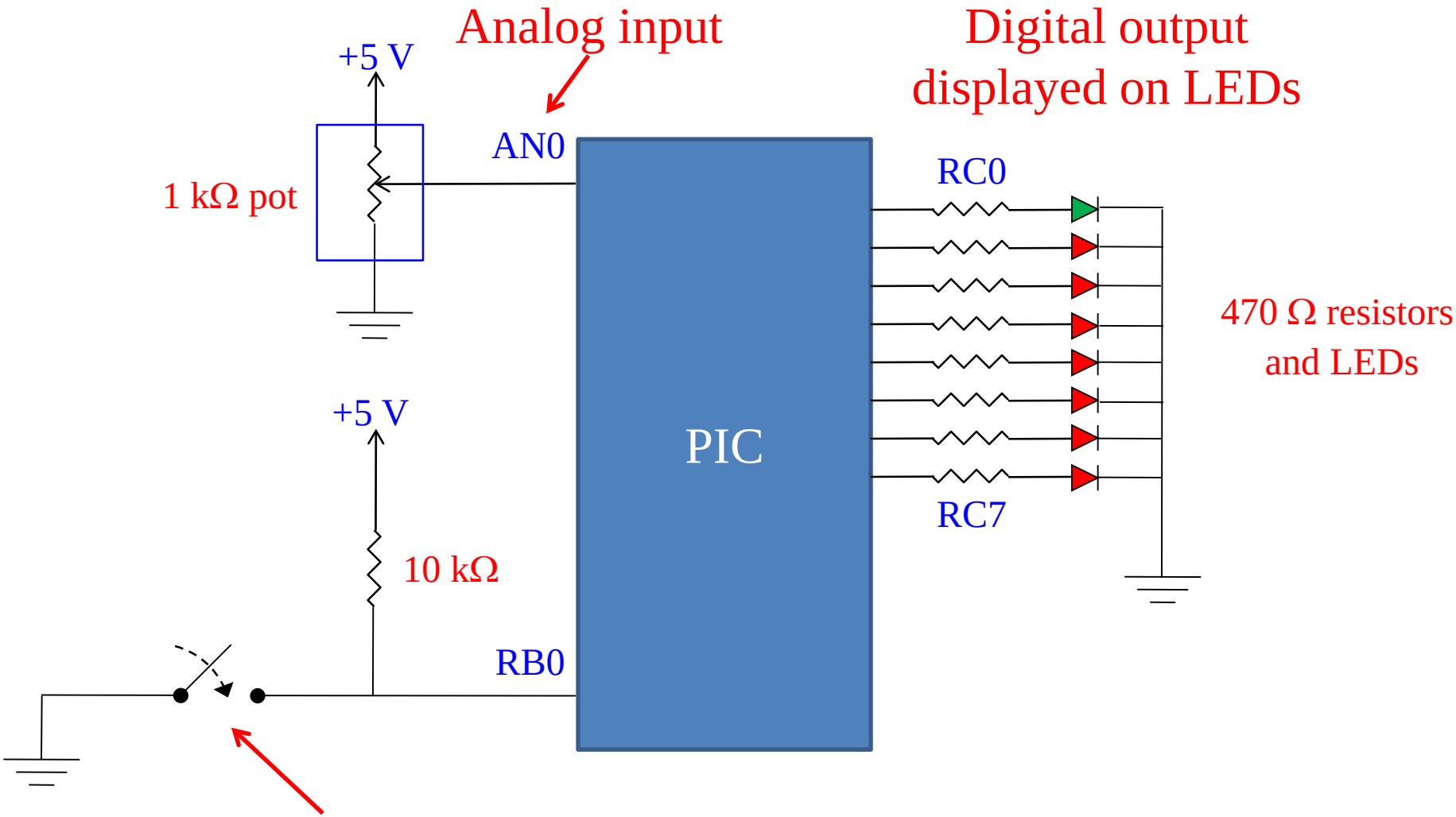
Rick Rarick

Summary 2018

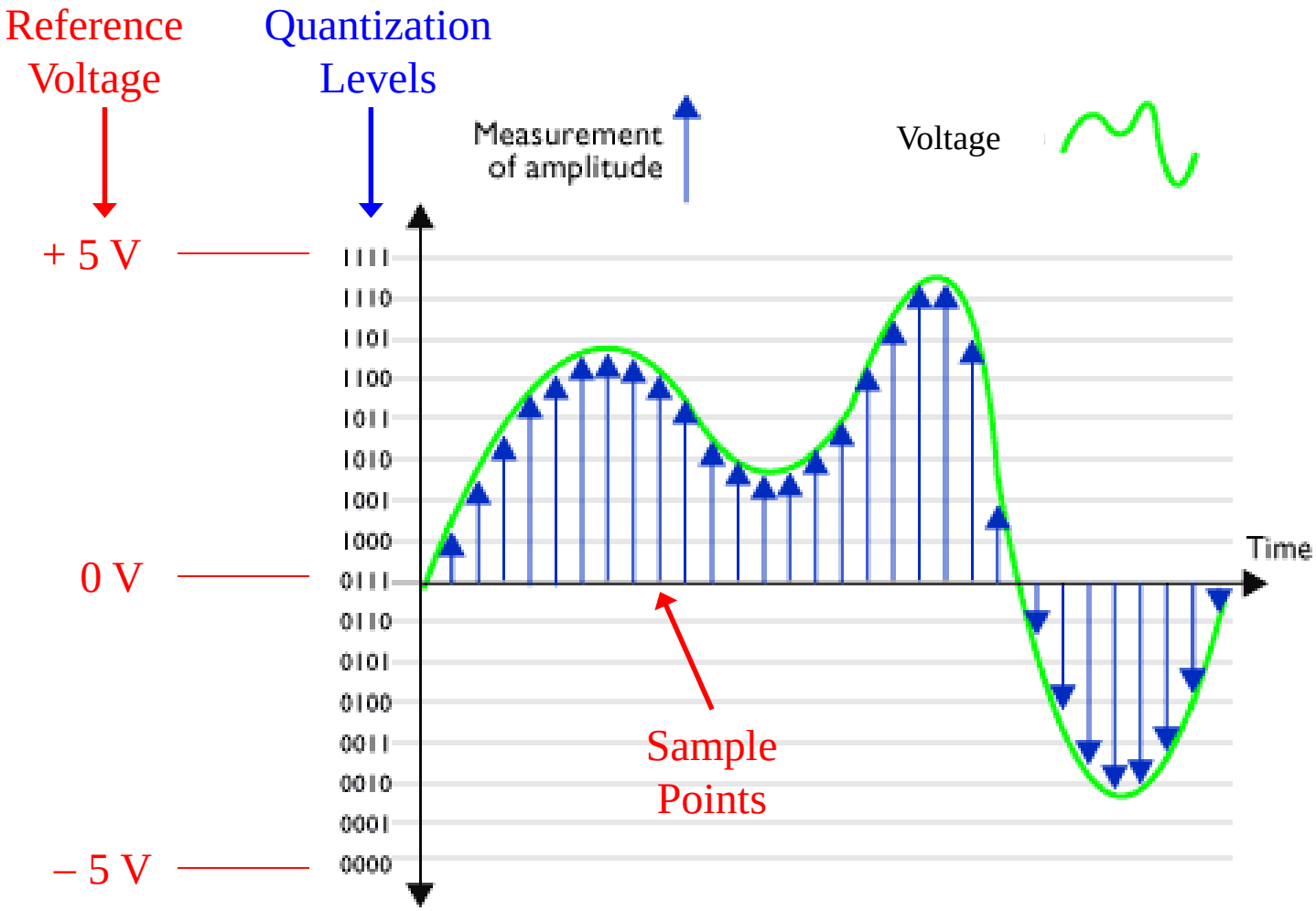
Lab 2 Outline

1. Analog-to-digital conversion
2. Analog-to-digital conversion module
3. Timer 0
4. Data register bank selection
5. Lab 2 hardware setup

Lab 2: Analog to Digital Conversion



Analog-to-digital converter (ADC)



Reference voltage must be supplied to the ADC

Analog-to-digital converter (ADC)

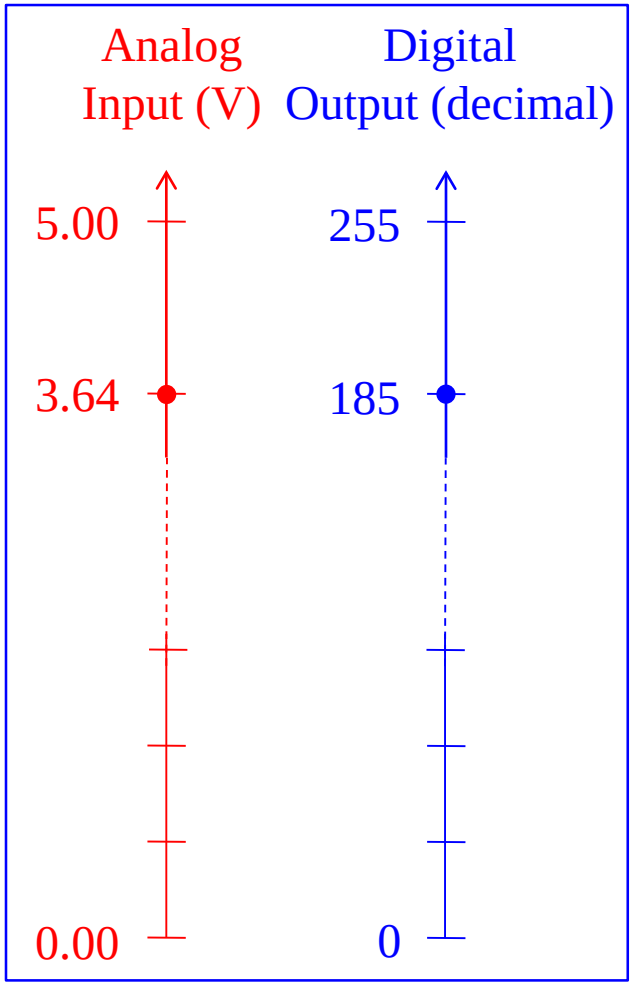
Example:

1. 0.00 to 5.00 volt reference
2. 8-bit digital representation
3. 3.64 V input

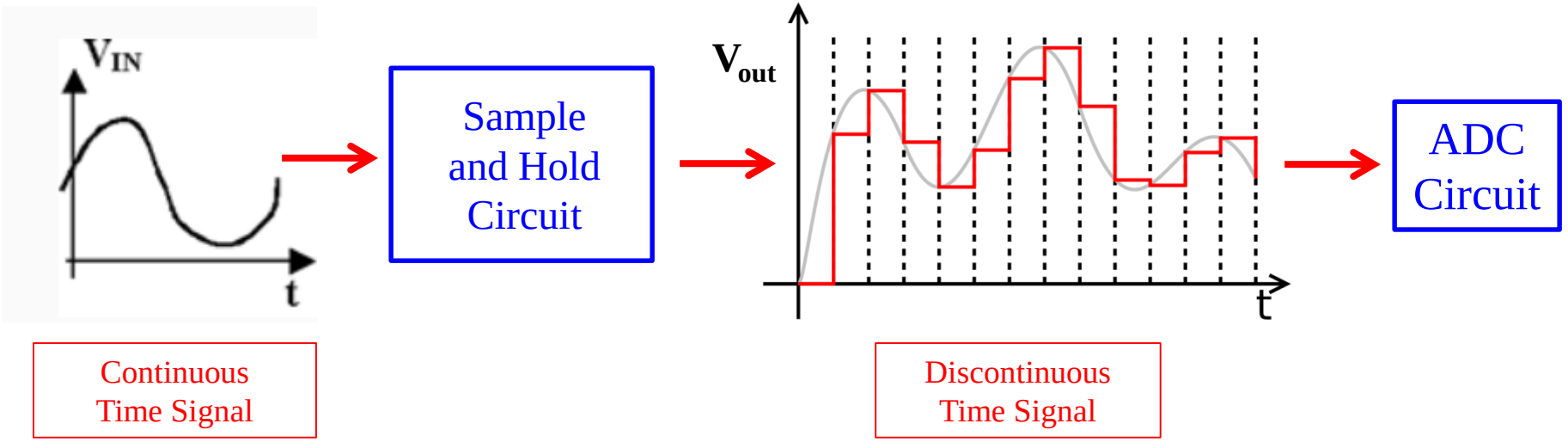
$$\frac{3.64}{5.00} \times 255 = 185.64$$

The PIC Truncates the 0.64

$$185 = B'10111001'$$



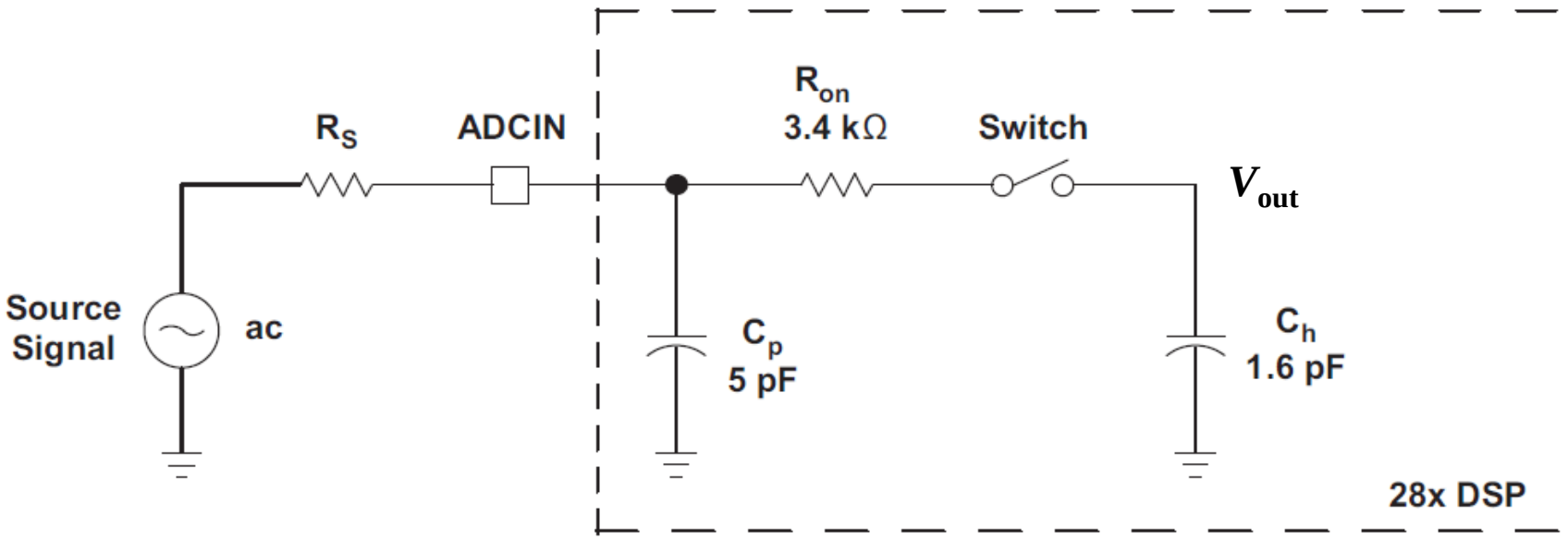
Sample and Hold Circuit



Function of S&H circuit is to capture the signal value at a given instant and hold it until the ADC can convert the voltage to a binary value .

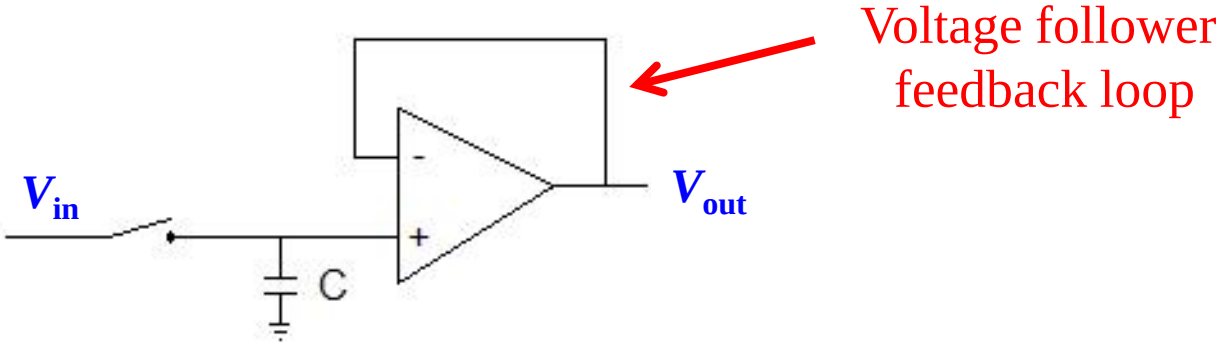
Sample and Hold Circuit

1. Many types of S&H circuits. Example: capacitor.
2. Sampling of input **ADCIN** begins when switch is closed.
3. Need a delay while the capacitor charges.
4. Sampling ends after capacitor is charged and switch opens.
5. Output V_{out} is held at **ADCIN** during A/D conversion.



Sample and Hold Circuit

1. Many types of S&H circuits. Example: voltage follower.
2. Sampling of input V_{in} begins when switch is closed.
3. Need a delay while the capacitor charges.
4. Sampling ends after capacitor is charged and switch opens.
5. Output V_{out} is held at V_{in} during A/D conversion.

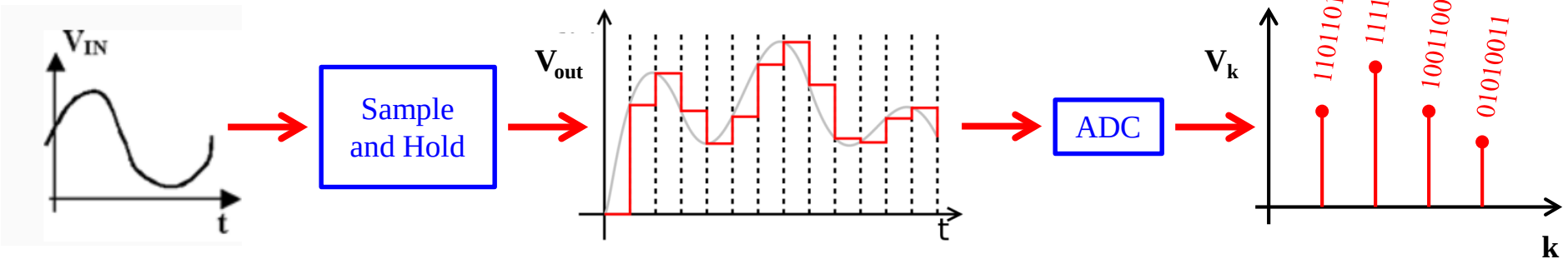


Opamp configured as a voltage follower

2-Step A/D Conversion Process

Step 1

Step 2



Continuous Time Signal

Discontinuous Time Signal

Discrete Time Signal

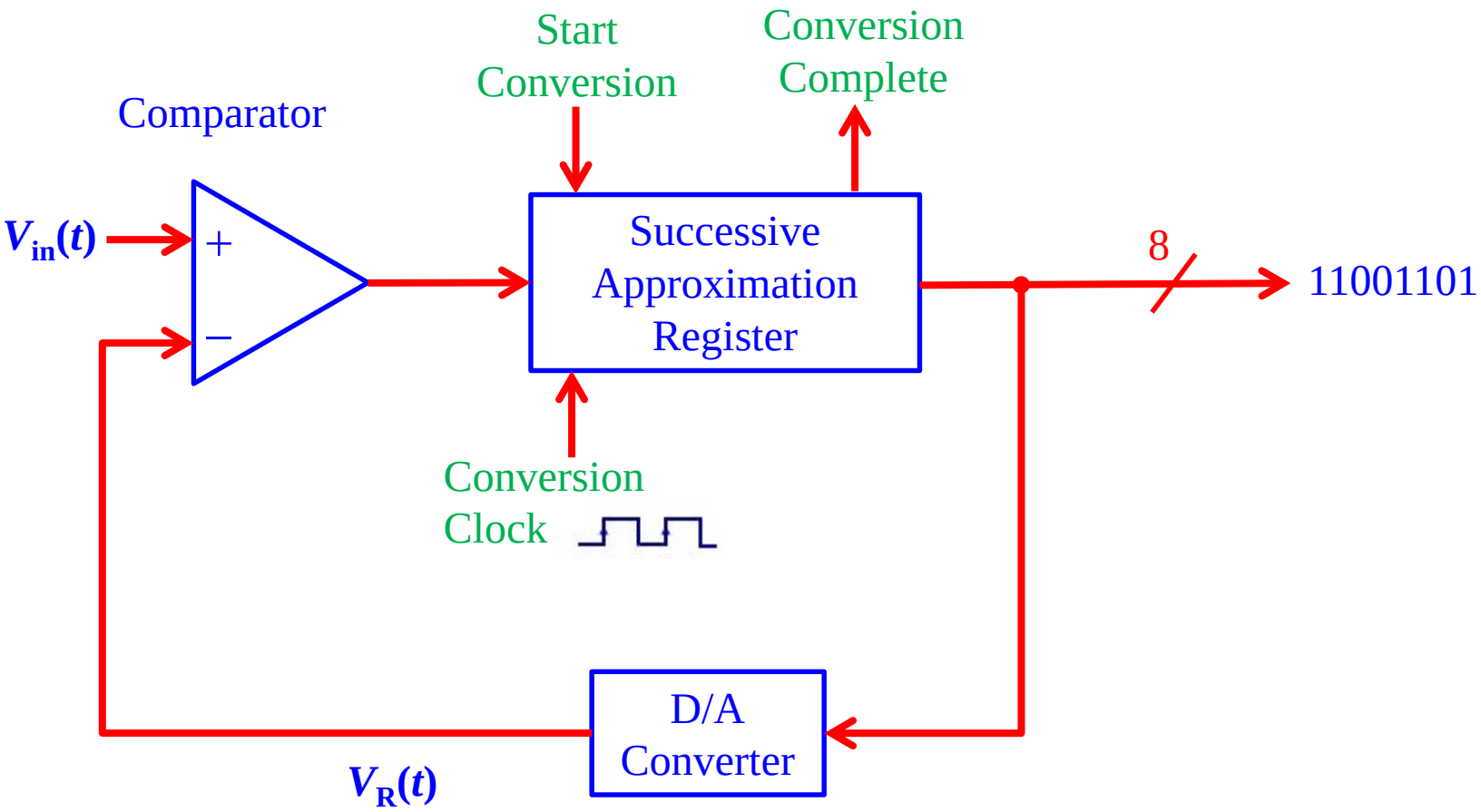
Analog to Digital Conversion (ADC)

1. Many types of ADCs.

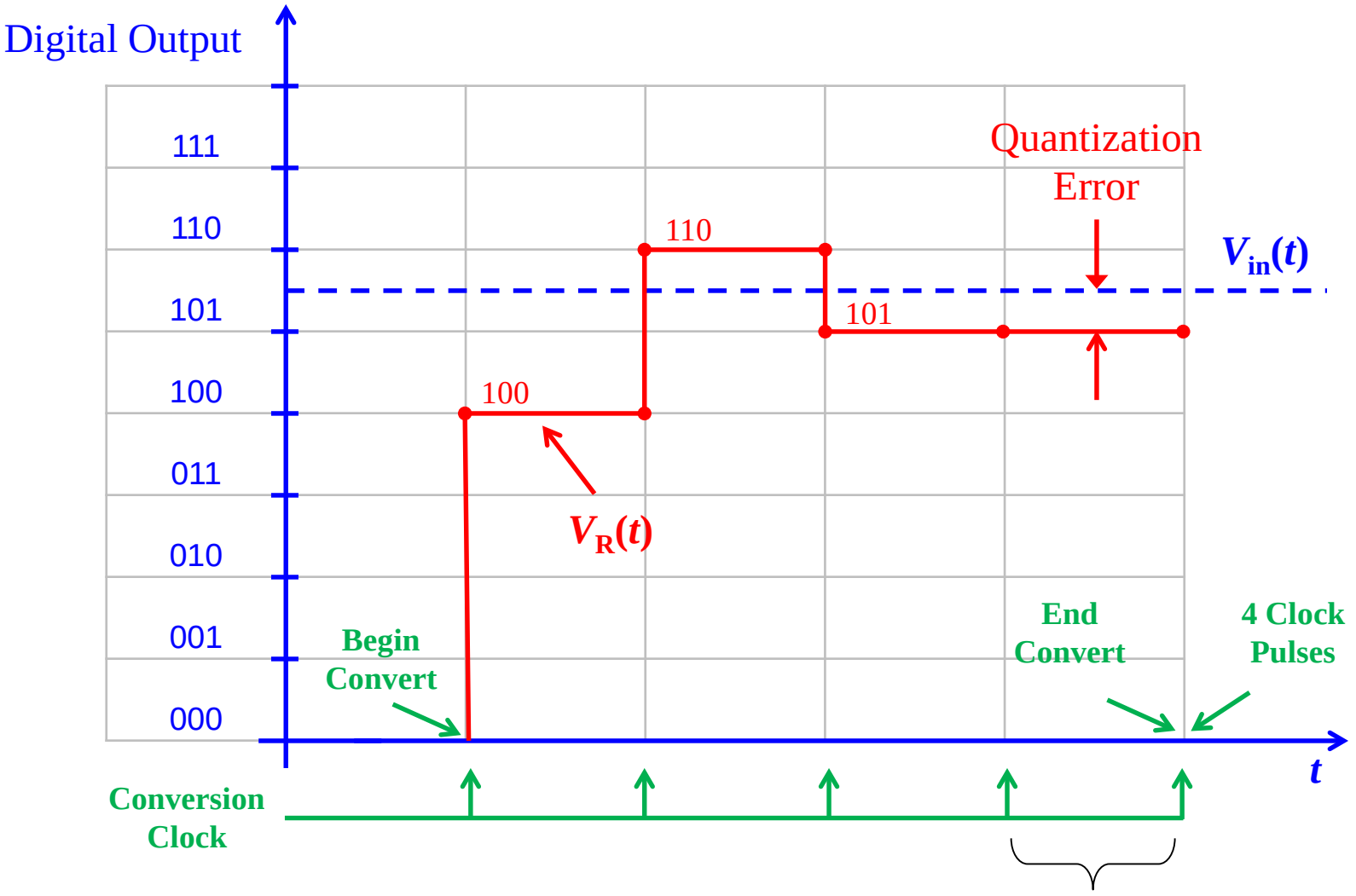
2. Basic example: Successive Approximation ADC.

3. Uses a binary search through all possible quantization levels and converges to a digital output for each conversion.

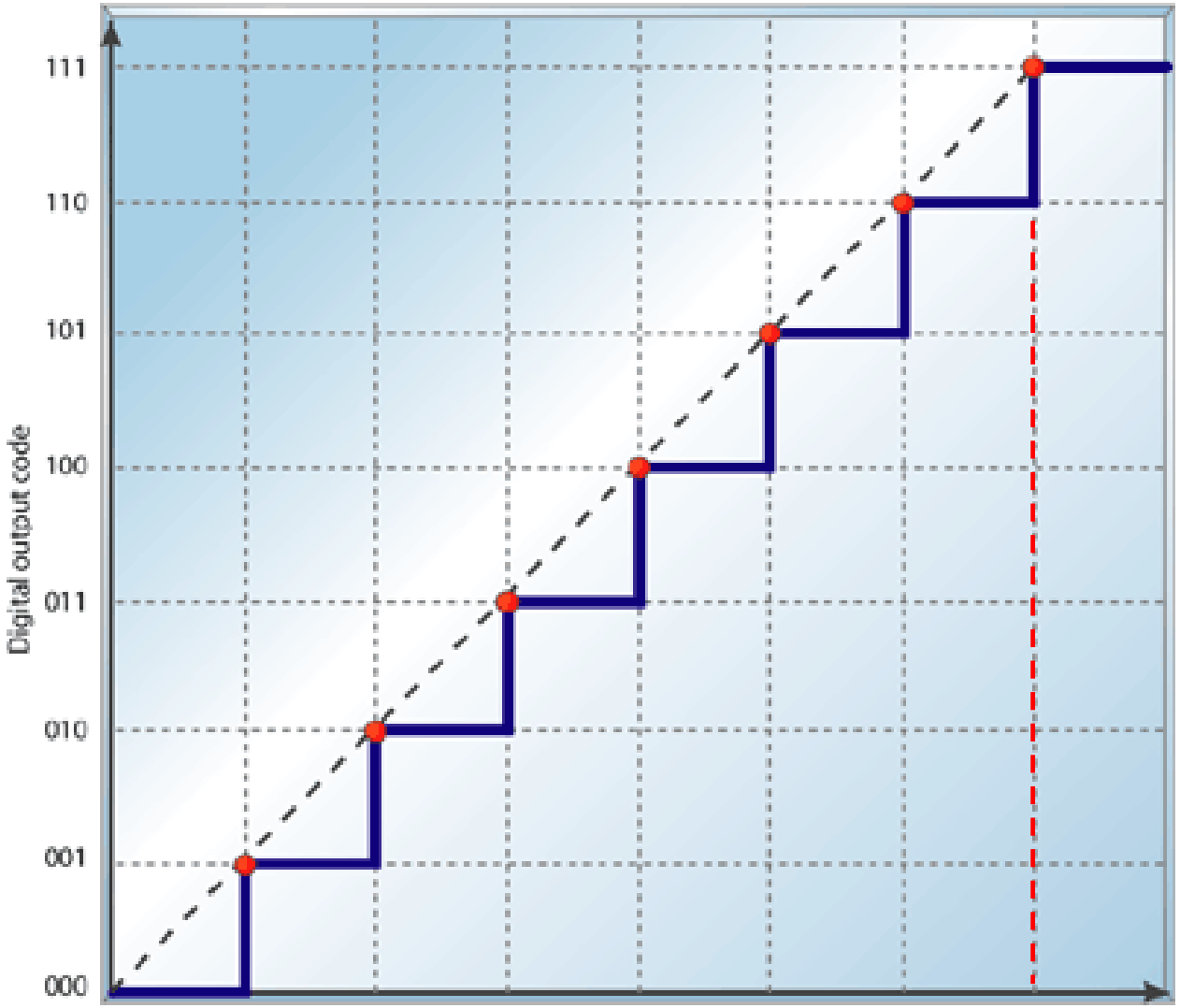
8-bit Successive Approximation ADC Block Diagram



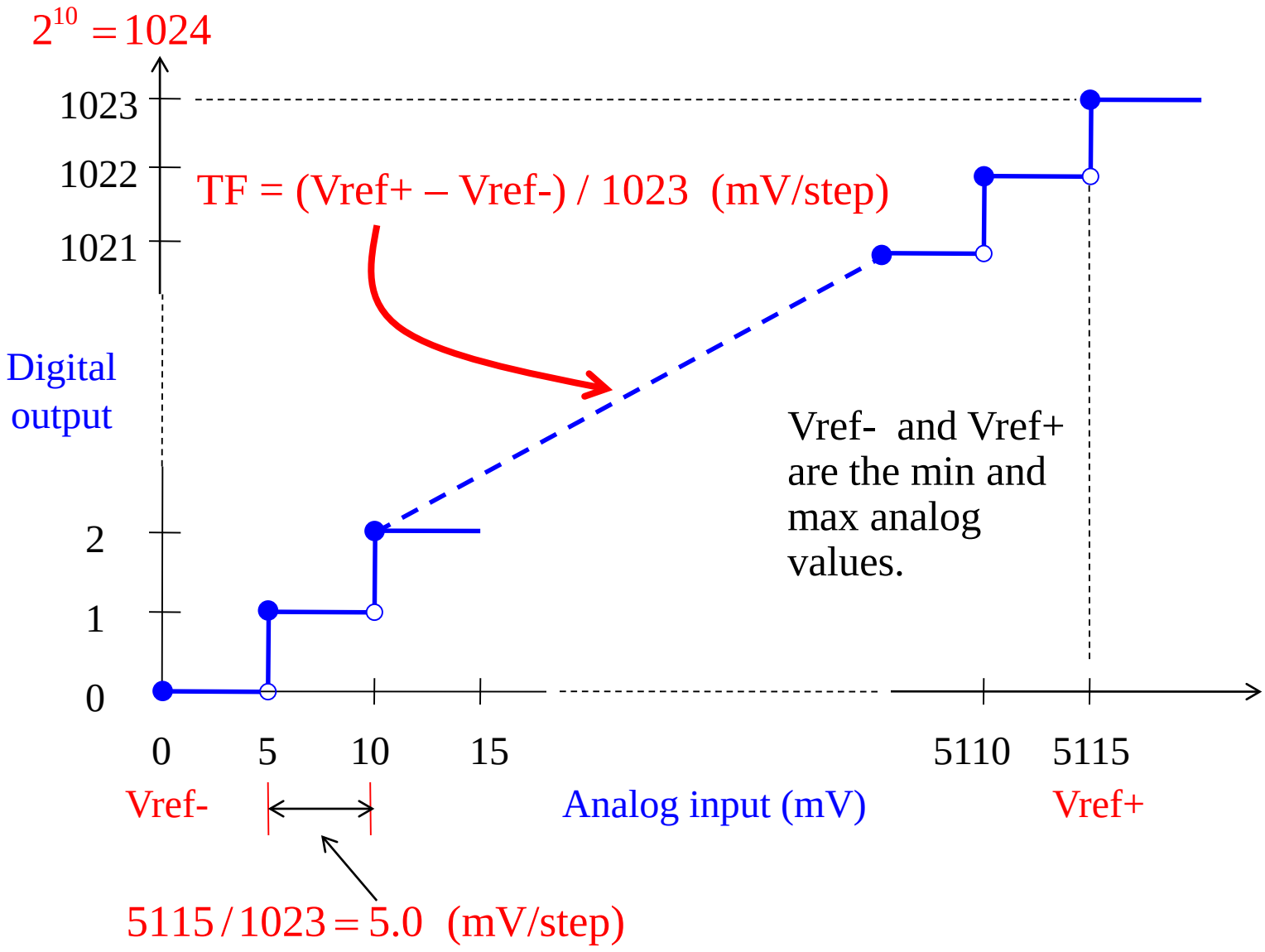
3-bit ADC Successive Approximation Logic with Truncation



3-bit ADC Transfer Function (with Truncation)



10-bit ADC Transfer Function (TF)



High-end ADC

Lab 2 Outline

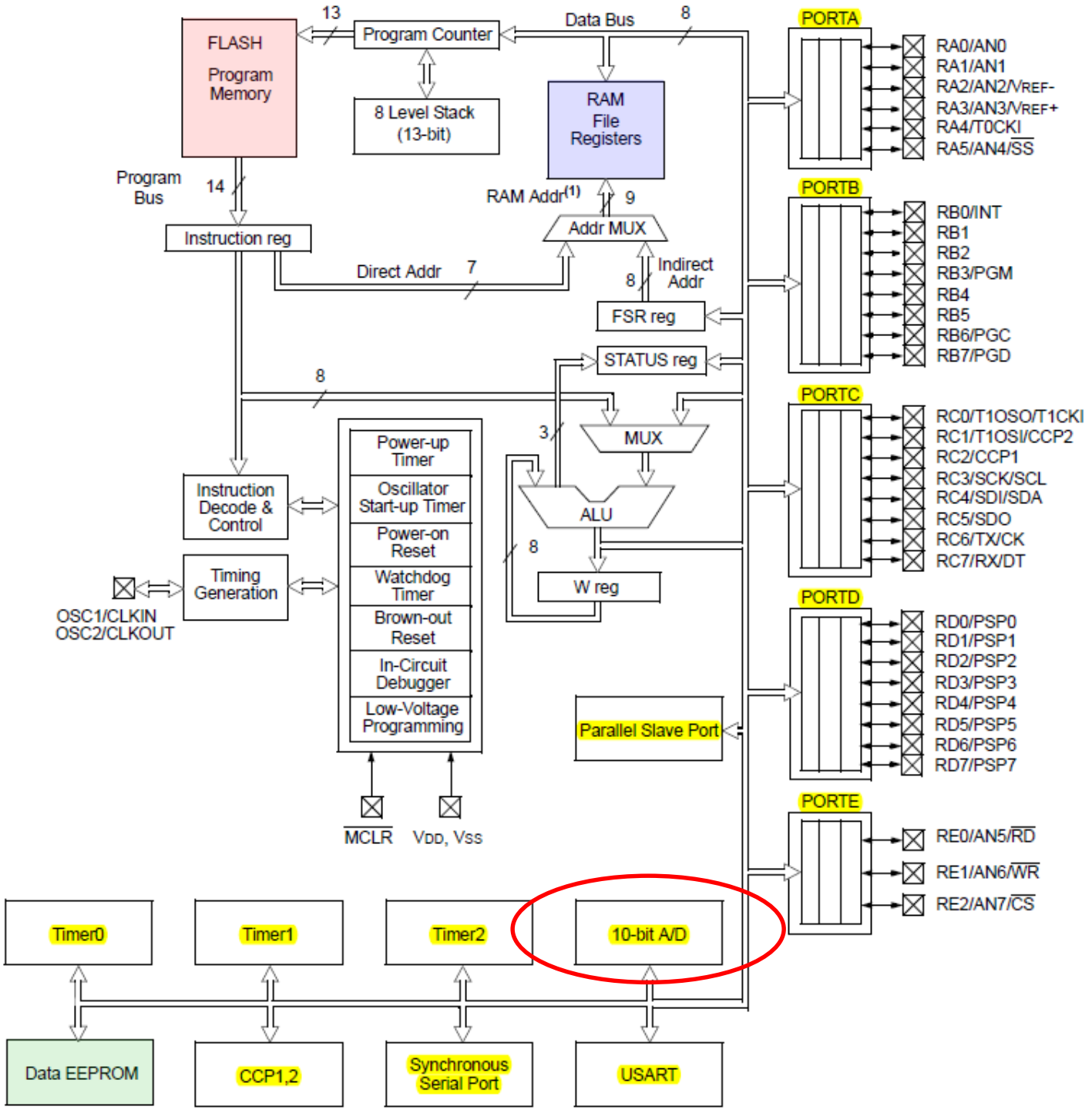
1. Analog-to-digital conversion
2. Analog-to-digital conversion module
3. Timer 0
4. Data register bank selection
5. Lab 2 hardware setup

PIC 16F877 Architecture

2^{13} = 8192 program memory addresses (Flash EEPROM)

2^{9} = 512 data memory addresses (SRAM)

2^{8} = 256 data memory addresses (EEPROM)



Data sheet p. 6

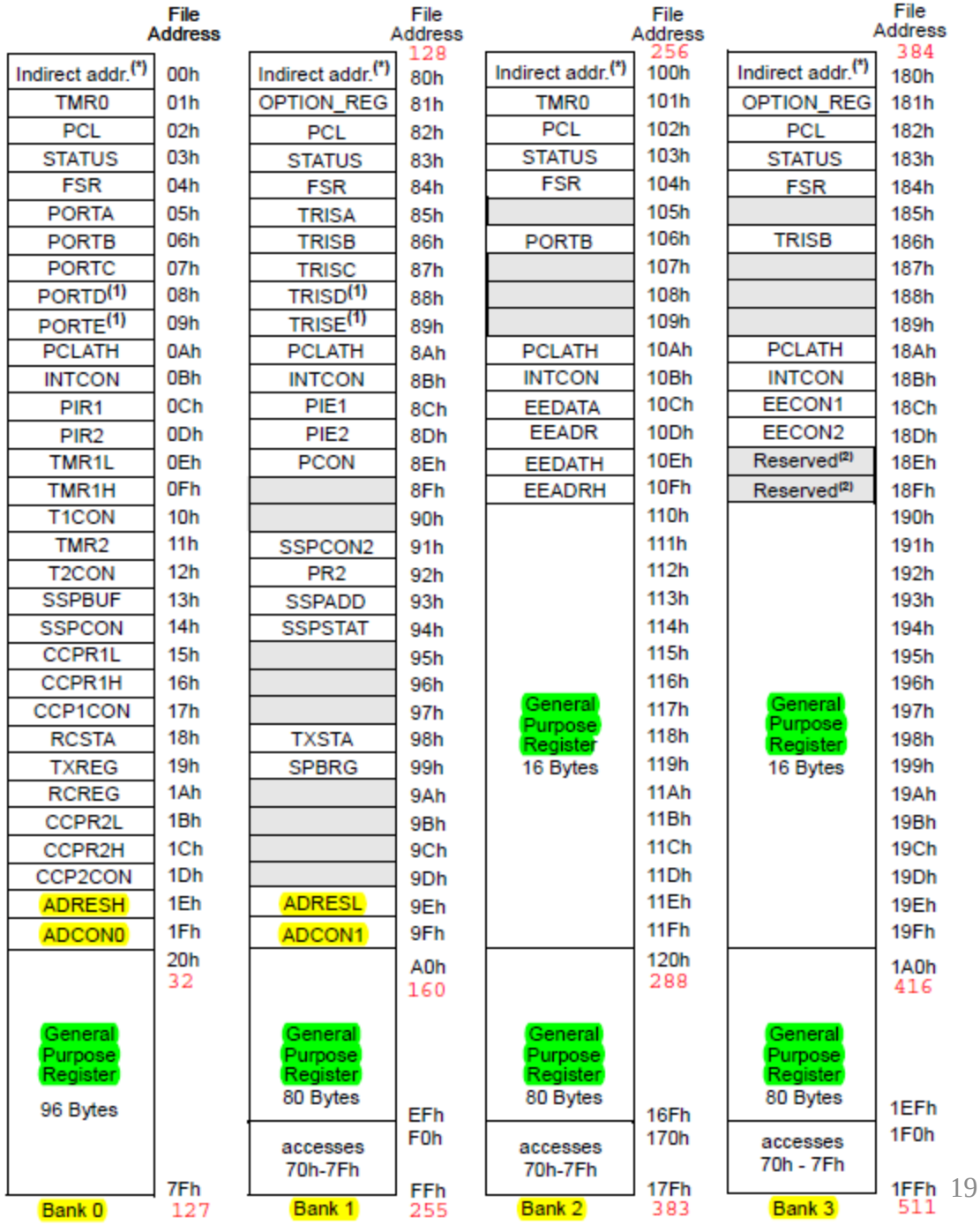
The Analog-to-Digital Converter Module (Peripheral)

There are four registers directly associated with the A/D module.

1. A/D Control Register0 (ADCON0)
2. A/D Control Register1 (ADCON1)
3. A/D Result High Register (ADRESH)
4. A/D Result Low Register (ADRESL)

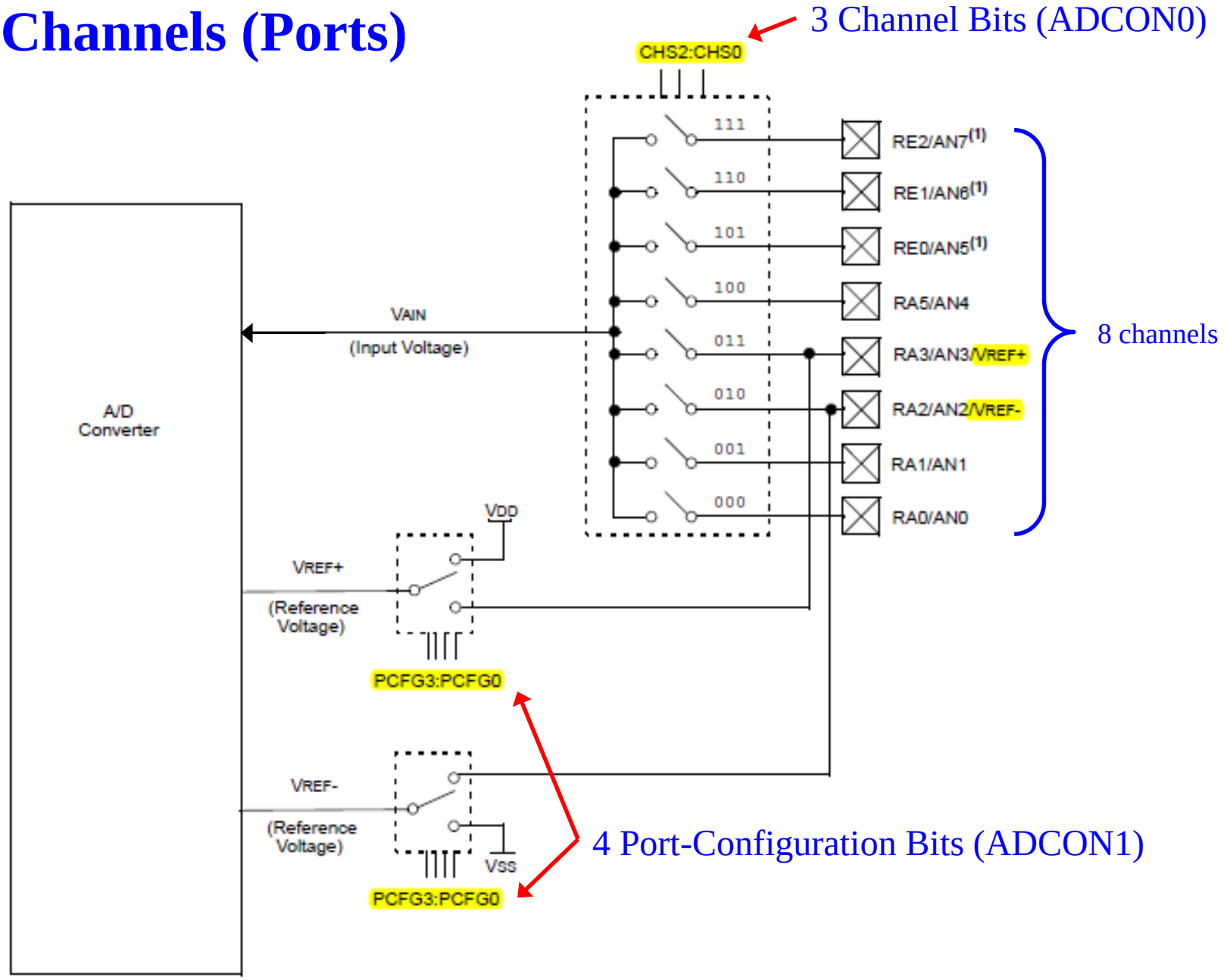
ADC Control Registers

Location in Data Memory



Page 13, data sheet

ADC Channels (Ports)



Page 113, Data sheet

ADCON0 Register for Lab 2

REGISTER 11-1: ADCON0 REGISTER (ADDRESS: 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON
bit 7							bit 0

bit 7-6 **ADCS1:ADCS0:** A/D Conversion Clock Select bits
00 = Fosc/2
01 = Fosc/8
10 = Fosc/32
11 = FRC (clock derived from the internal A/D module RC oscillator)

bit 5-3 **CHS2:CHS0**: Analog Channel Select bits
000 = channel 0, (RA0/AN0)
001 = channel 1, (RA1/AN1)
010 = channel 2, (RA2/AN2)
011 = channel 3, (RA3/AN3)
100 = channel 4, (RA5/AN4)
101 = channel 5, (RE0/AN5)(1)
110 = channel 6, (RE1/AN6)(1)
111 = channel 7, (RE2/AN7)(1)

Lab 2 uses AN0 for analog input and Fosc / 8.

Fosc: Oscillator frequency

bit 2 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
If ADON = 1:
1 = A/D conversion in progress (setting this bit starts the A/D conversion)
0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)

bit 1 **Unimplemented**: Read as '0'

bit 0 **ADON**: A/D On bit
1 = A/D converter module is operating
0 = A/D converter module is shut-off and consumes no operating current

Instructions:

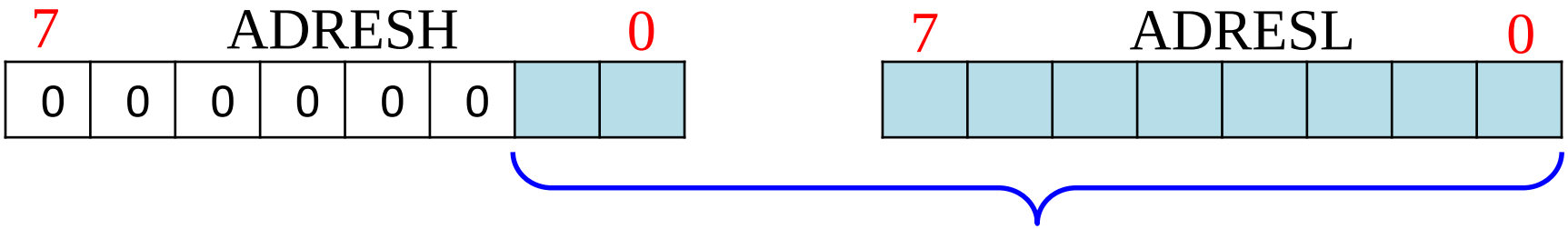
```
movlw     B'01000001'          ; Fosc/8, AN0, A/D enabled
movwf     ADCON0               ; ADCON1 = 01000001b in C
```

A/D Result Registers

The ADC result is 10 bits, stored in ADRESH:ADRESL.

Justification is controlled by the ADFM bit of ADCON1.

ADFM: "A/D Format"



10 bits, right justified (ADFM = 1)

10 bits, left justified (ADFM = 0)

ADCON1: A/D Result Format

REGISTER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)

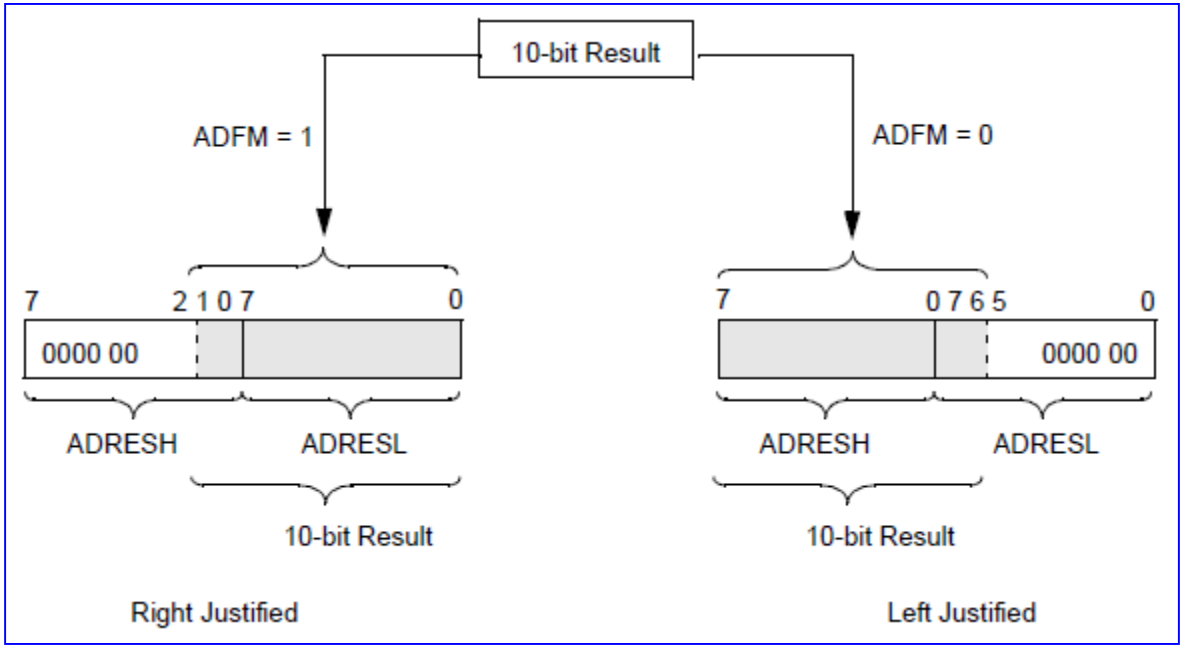
U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'.
0 = Left justified. 6 Least Significant bits of ADRESL are read as '0'.

bit 6-4 **Unimplemented:** Read as '0'

bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits:

Page 112, data sheet



Page 116, data sheet

ADCON1: A/D Port Configuration

REGISTER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'.
0 = Left justified. 6 Least Significant bits of ADRESL are read as '0'.

bit 6-4 **Unimplemented:** Read as '0'

bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits:

PCFG3: PCFG0	AN7[1] RE2	AN6[1] RE1	AN5[1] RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN/ Refs[2]
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

A = Analog input D = Digital I/O

number of analog channels available as A/D inputs / the number of analog channels used as voltage reference inputs.

For Lab 2

Datasheet, p. 112

ADCON1 - A/D Port Configuration Control

REGISTER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'.
0 = Left justified. 6 Least Significant bits of ADRESL are read as '0'.

bit 6-4 **Unimplemented:** Read as '0'

bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits:

Instructions:

```
movlw   B'00001110'  ; Left justify: ADFM = 0
                     ; 1 analog channel (AN0),
                     ; VDD and VSS references:
                     ; <PCFG3:PCFG0> = 1110
movwf   ADCON1
```

ADC Module - Associated Registers

TABLE 11-2: REGISTERS/BITS ASSOCIATED WITH A/D

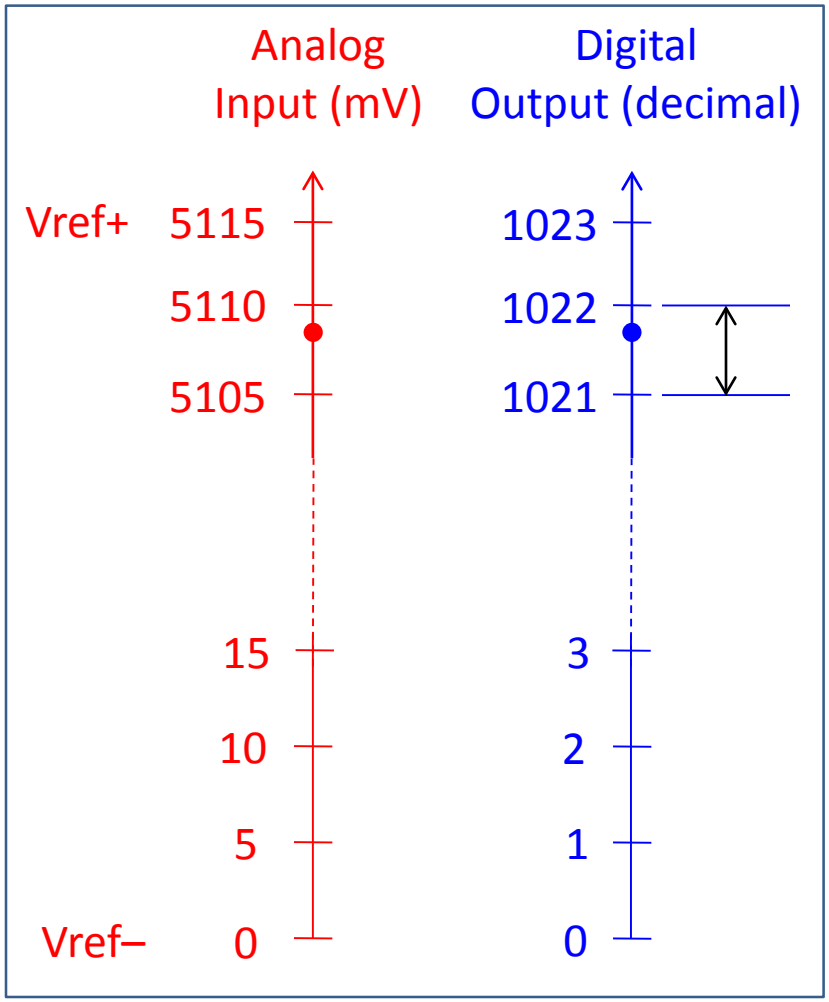
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on $\overline{MCLR}$, WDT
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	—	ADON	0000 00-0	0000 00-0
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	--0u 0000
89h(1)	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction bits			0000 -111	0000 -111
09h(1)	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used for A/D conversion.

Note 1: These registers/bits are not available on the 28-pin devices.

Page 117, Data sheet

10-bit A/D Resolution @ [0 to 5.115] V Range



$$\text{Transfer Function} = \frac{5115\ \text{mV}}{1023\ \text{steps}} = 5.0\ \frac{\text{mV}}{\text{step}}$$

Assume Truncation

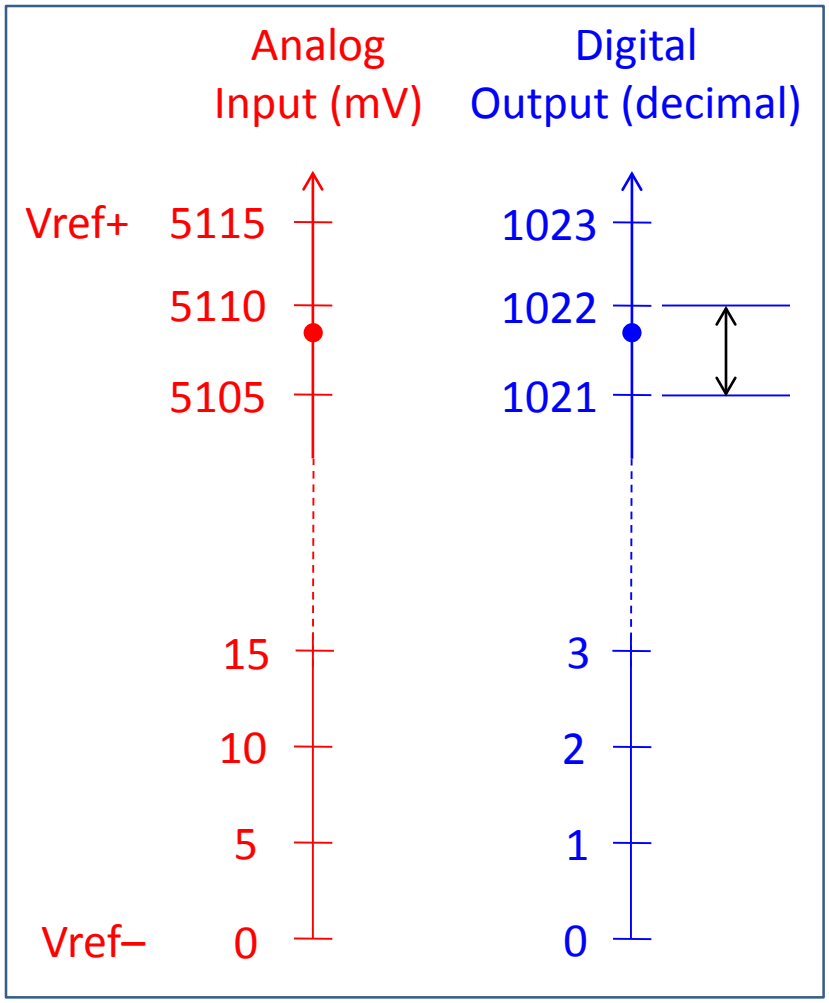
$$\text{Resolution} = 1\ \text{step} = 5.0\ \text{mV}$$

$$\text{Resolution Percent} = \frac{1\ \text{step}}{1023\ \text{steps}}(100)$$

$$\approx 0.098\%$$

- "Resolution" is also called "Quantization Interval."
- For truncation ADCs, Quantization Interval = Maximum Quantization Error

10-bit Analog to Digital



$$\text{Resolution} = \frac{(V_{\text{ref}}+) - (V_{\text{ref}}-)}{2^n - 1} \quad \frac{\text{volts}}{\text{step}}$$

Analog to Digital Conversion
(with truncation):

A = analog value (mV)
D = digital value (decimal)

$$D = \left\lfloor \frac{\left(2^n - 1\right)\left(A - (V_{ref}-)\right)}{(V_{ref}+) - (V_{ref}-)} \right\rfloor$$

Ex: $A = 5108,\ (V_{ref}+) = 5115,\ (V_{ref}-) = 0$ [mV]

$$D = \left\lfloor \frac{1023(5108 - 0)}{5115 - 0} \right\rfloor = \lfloor 1021.6 \rfloor = 1021$$

10-bit Digital to Analog

Given an *n*-bit ADC with a digital output *D* (in decimal), what range of analog inputs corresponds to this? (Assume truncation.)

Example: 10-bit ADC, $\text{Vref+} = 5115 \text{ mV}, \quad \text{Vref}- = 0, \quad D = 1021$

Answer:

$$\text{Analog Input Range (mV)} = (D \ + \ 1/2 \ \pm 1/2)\frac{(\text{Vref}+)-(\text{Vref}-)}{2^n - 1}$$

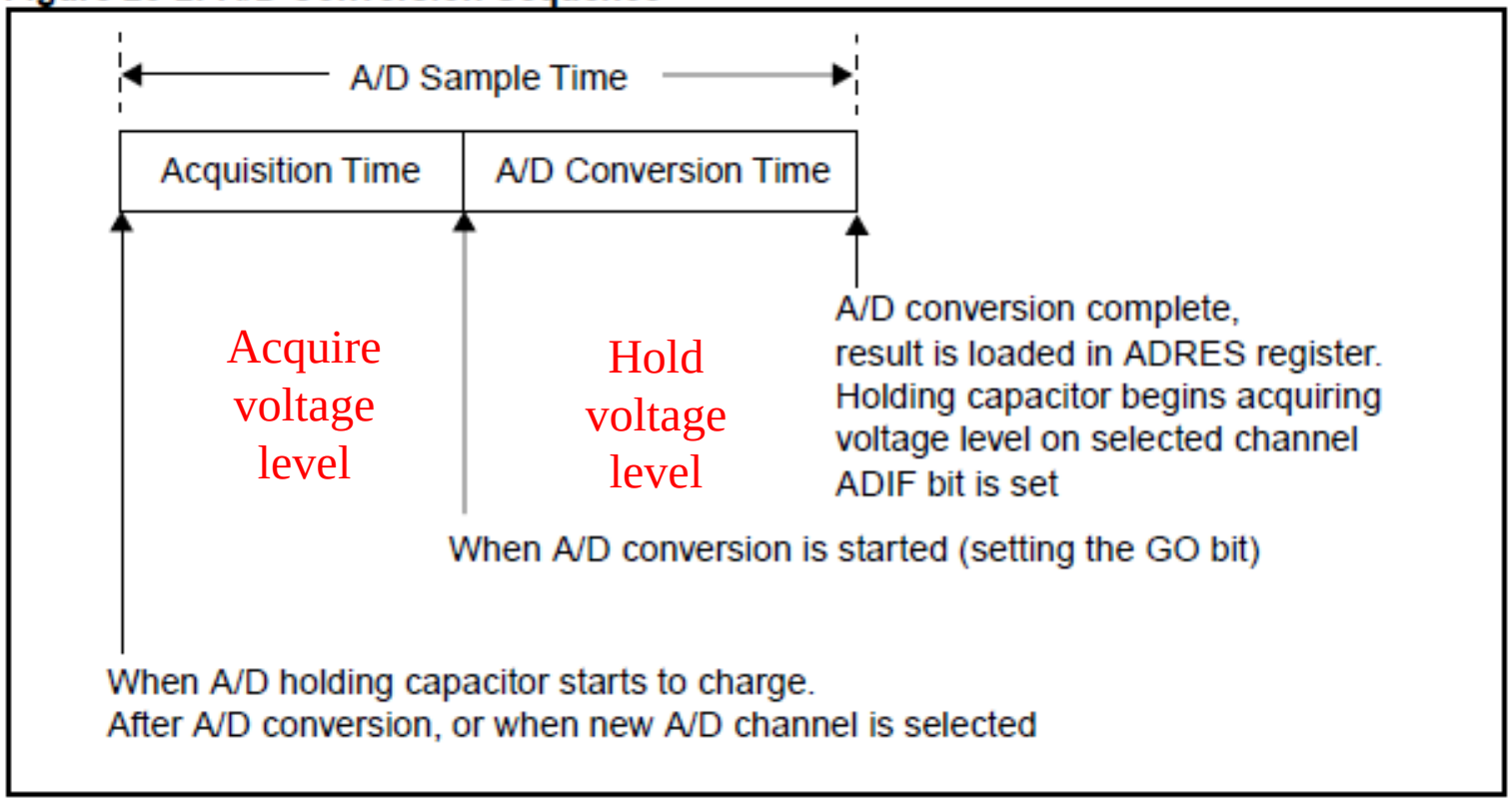
$$= (1021 \ + \ 1/2 \ \pm 1/2)\frac{5115-0}{2^{10}-1}$$

$$= (1021.5 \pm 0.5)(5)$$

$$= 5107.5 \pm 2.5 \ \text{[mV]}$$

ADC Timing

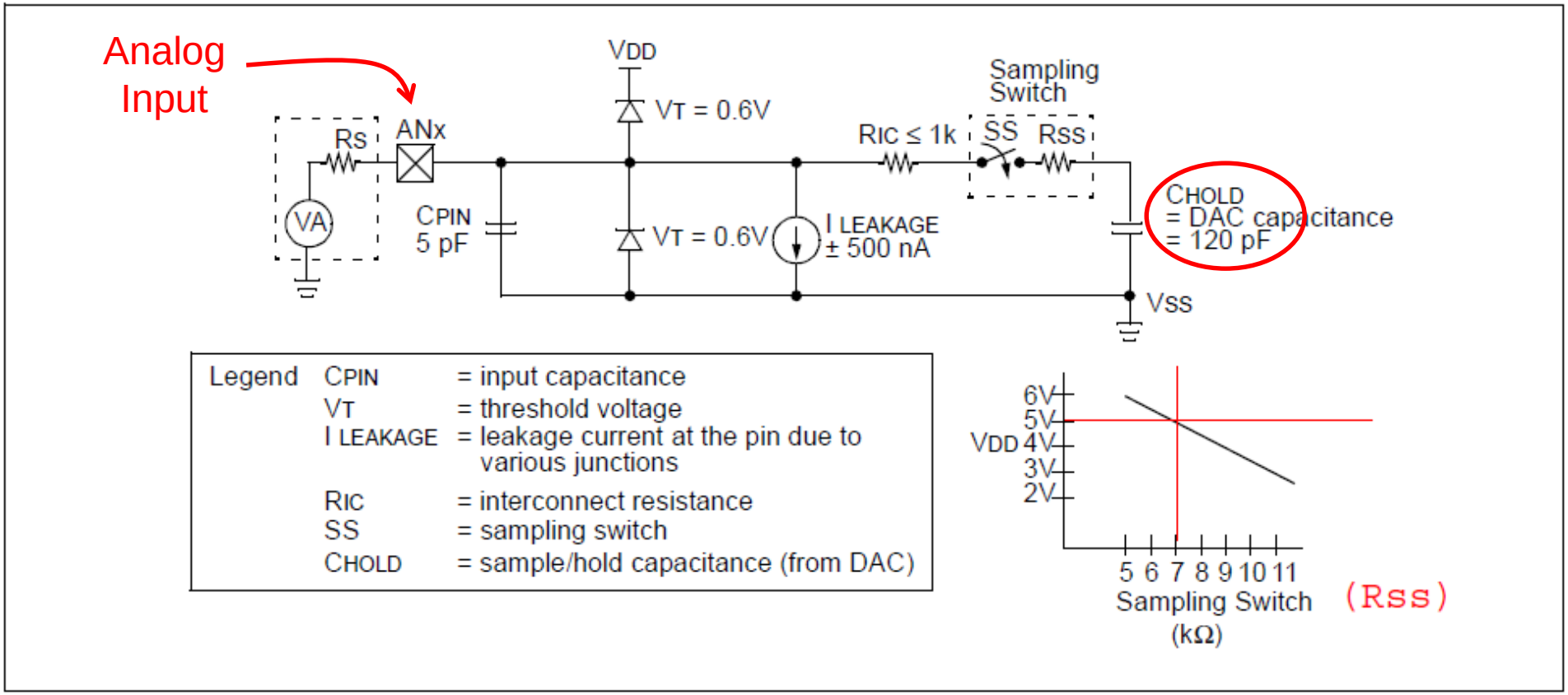
Figure 23-2: A/D Conversion Sequence



Page 23-5, Mid-Range Reference Manual

PIC Sample/Hold Circuit

FIGURE 11-2: ANALOG INPUT MODEL



Data sheet p. 114

ADC Acquisition Time: T_{ACQ}

T_{ACQ} = Analog signal acquisition time

= Amplifier settling time

+ Capacitor charge time

+ Temperature coefficient time

$= T_{AMP} + T_C + T_{COFF}$

For a typical system operating at 50° C

$$\boxed{T_{ACQ} = 19.72\ \mu s}$$

Page 23-6, Mid-Range Reference Manual

Selecting the A/D Conversion Clock Period T_{AD}

1. Conversion clock period = T_{AD}
   Conversion clock frequency = $F_{AD} = 1 / T_{AD}$

2. For correct A/D conversions, $T_{AD} \geq 1.6\ \mu s$ (datasheet)

3. Configure the conversion clock with ADCON0 register

REGISTER 11-1: ADCON0 REGISTER (ADDRESS: 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON
bit 7							bit 0

bit 7-6 **ADCS1:ADCS0:** A/D Conversion Clock Select bits
00 = Fosc/2
01 = Fosc/8
10 = Fosc/32
11 = FRC (clock derived from the internal A/D module RC oscillator)

Pages 111 – 115 Data sheet

A/D Conversion Clock Period: TAD

For the 16F877, choose ADCON0<ADSC1:ADSC0> = 01, that is, choose

$$F_{\text{AD}} = \frac{F_{\text{OSC}}}{8}$$

because

$$T_{\text{AD}} = \frac{1}{F_{\text{AD}}} = \frac{8}{F_{\text{OSC}}} = \frac{8}{3.6864\ (\text{MHz})} = 2.\ 2\mu s$$

$$2.\ 2\mu s\ \geq 1.6\ \mu s$$

Page 111, Data sheet

Starting the A/D Conversion

A/D conversion is started by setting the "GO" bit in the ADCON0 register

ADCON0 REGISTER (ADDRESS: 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON
bit 7							bit 0

bit 2 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit

If ADON = 1:

1 = A/D conversion in progress (setting this bit starts the A/D conversion)

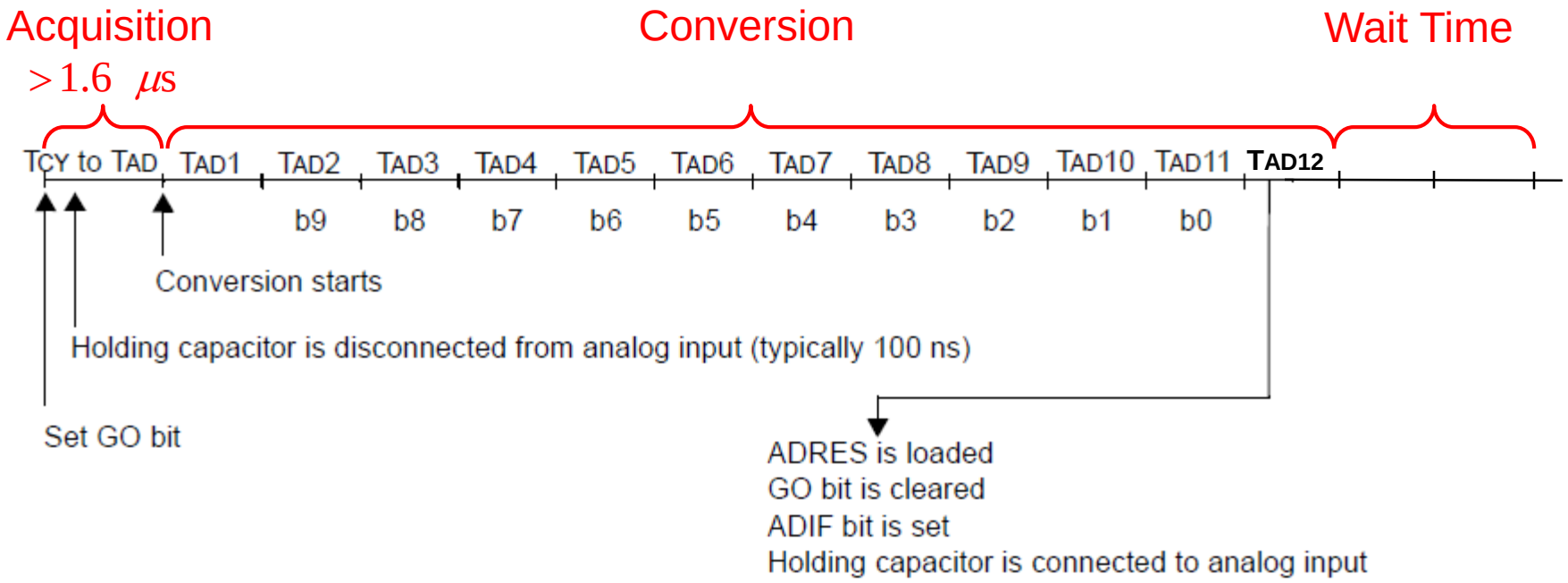
0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)

Instruction:

```
bsf  ADCON0, GO
```

Conversion Timing

A/D conversion clock cycles



12 T_{AD} periods are required for a conversion
plus 2 T_{AD} periods wait time before next acquisition.

Page 116, data sheet

ADC Total Sample Time

ADC sample time = $T_{ACQ} + T_{CONVERT} + T_{WAIT}$

$= T_{ACQ} + 12\ T_{AD} + 2\ T_{AD}$

$= 19.7\ \mu s + 14(2.2)\ \mu s$

$= 50.5\ \mu s$

ADC sample frequency = $1 / (50.5\ \mu s)$

$= 19.8$ kHz

A/D Interrupt Flag

1. `ADIF`: A/D Interrupt Flag

2. `PIR1`: Peripheral Interrupt Register 1

3. After the 12 T_{AD} periods, `ADIF` is automatically set by the hardware.

REGISTER 2-5: PIR1 REGISTER (ADDRESS 0Ch)

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

bit 7 **PSPIF(1):** Parallel Slave Port Read/Write Interrupt Flag bit
1 = A read or a write operation has taken place (must be cleared in software)
0 = No read or write has occurred

bit 6 **ADIF:** A/D Converter Interrupt Flag bit
1 = An A/D conversion completed
0 = The A/D conversion is not complete

Datasheet p. 22

Lab 2 Outline

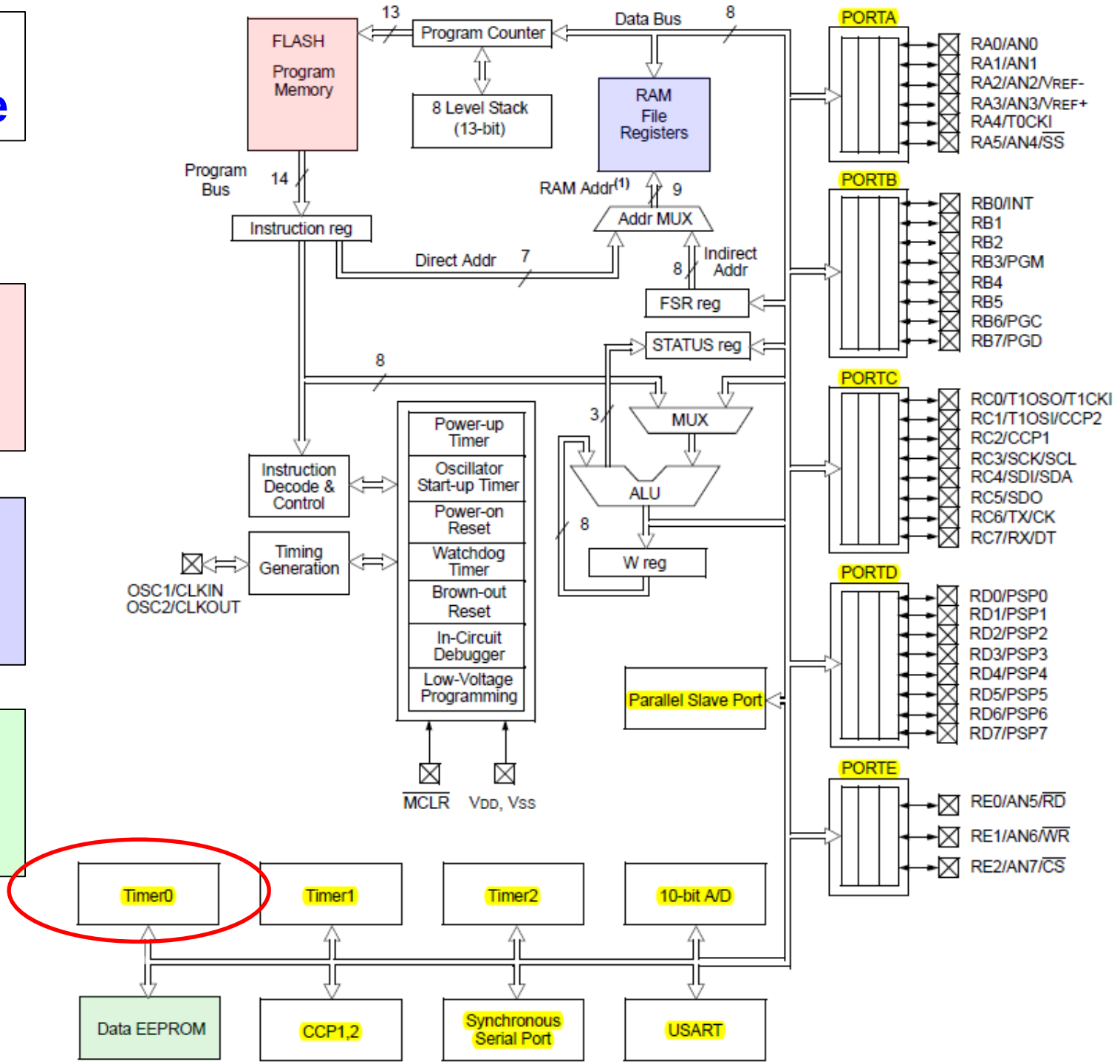
1. Analog-to-digital conversion
2. Analog-to-digital conversion module
3. Timer 0
4. Data register bank selection
5. Lab 2 hardware setup

PIC 16F877 Architecture

2^{13} = 8192 program memory addresses (Flash EEPROM)

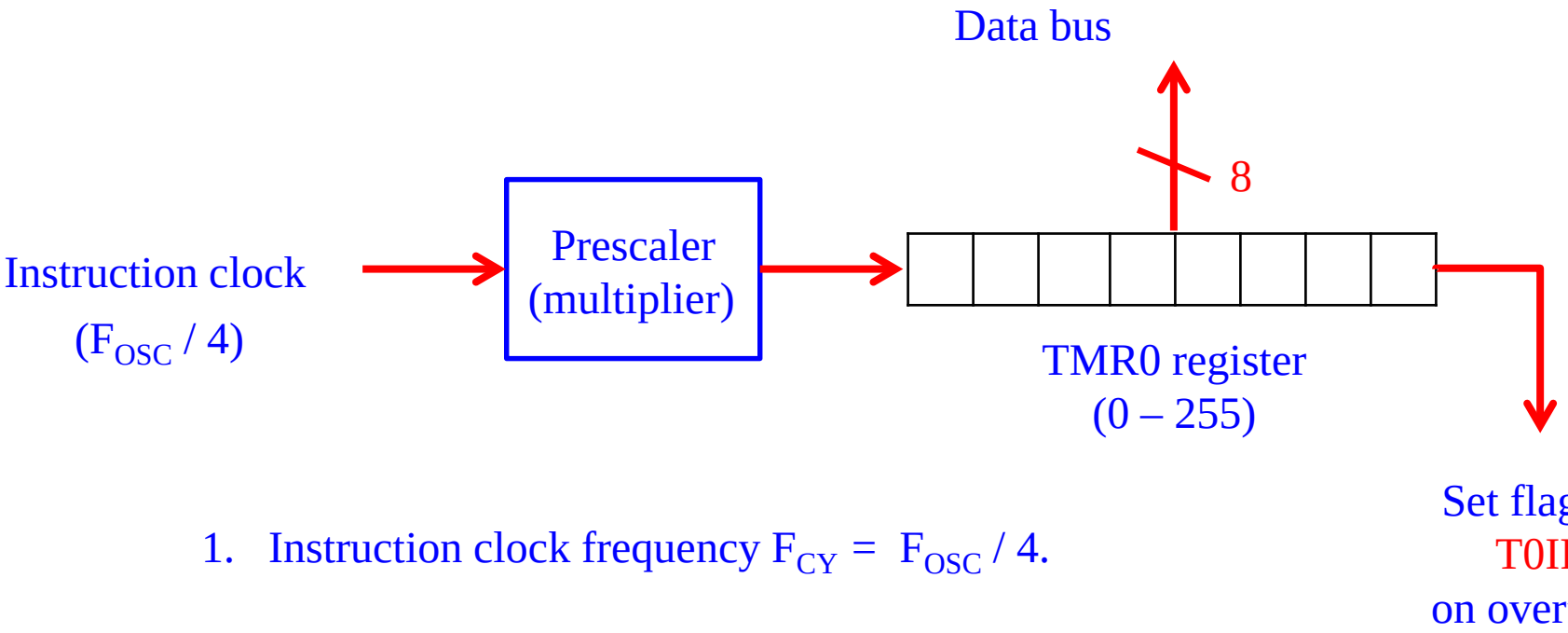
2^{9} = 512 data memory addresses (SRAM)

2^{8} = 256 data memory addresses (EEPROM)



Data sheet p. 6

Timer0 Module



1. Instruction clock frequency $F_{CY} = F_{OSC} / 4$.
2. TMR0 increments once every *Prescale* instructions.
3. T0IF = Timer0 Interrupt Flag bit in INTCON register.
4. T0IF set on overflow from 255 → 0
5. Timer0 interrupt flag set even if interrupt not enabled.

Timer0 Rollover Time

- Since the TMR0 register increments once every *Prescale* instructions, it takes (*Prescale* × 256) instruction cycles for TMR0 to rollover (overflow).

- Time for each instruction clock "tick":

$$\text{Instruction Period} = T_{CY} = \frac{4}{F_{OSC}} = 4T_{OSC} \quad \text{(sec)}$$

- Time for each Timer0 "tick":

$$\begin{aligned} \text{TMR0 Period} &= T_{TMR0} = \text{instruction period} \times Prescale \\ &= T_{CY} \times Prescale \\ &= \frac{4}{F_{OSC}} \times Prescale \end{aligned}$$

Timer0 Rollover Time

$$\text{Rollover Time} = \frac{4}{F_{OSC}} \times Prescale \times 256$$

1. If *Prescale* = 1 and F_{OSC} = 3.6864 MHz,

$$\text{Rollover Time} = 1.0851 \times 10^{-6} \times 1 \times 256 = 278\ \mu s$$

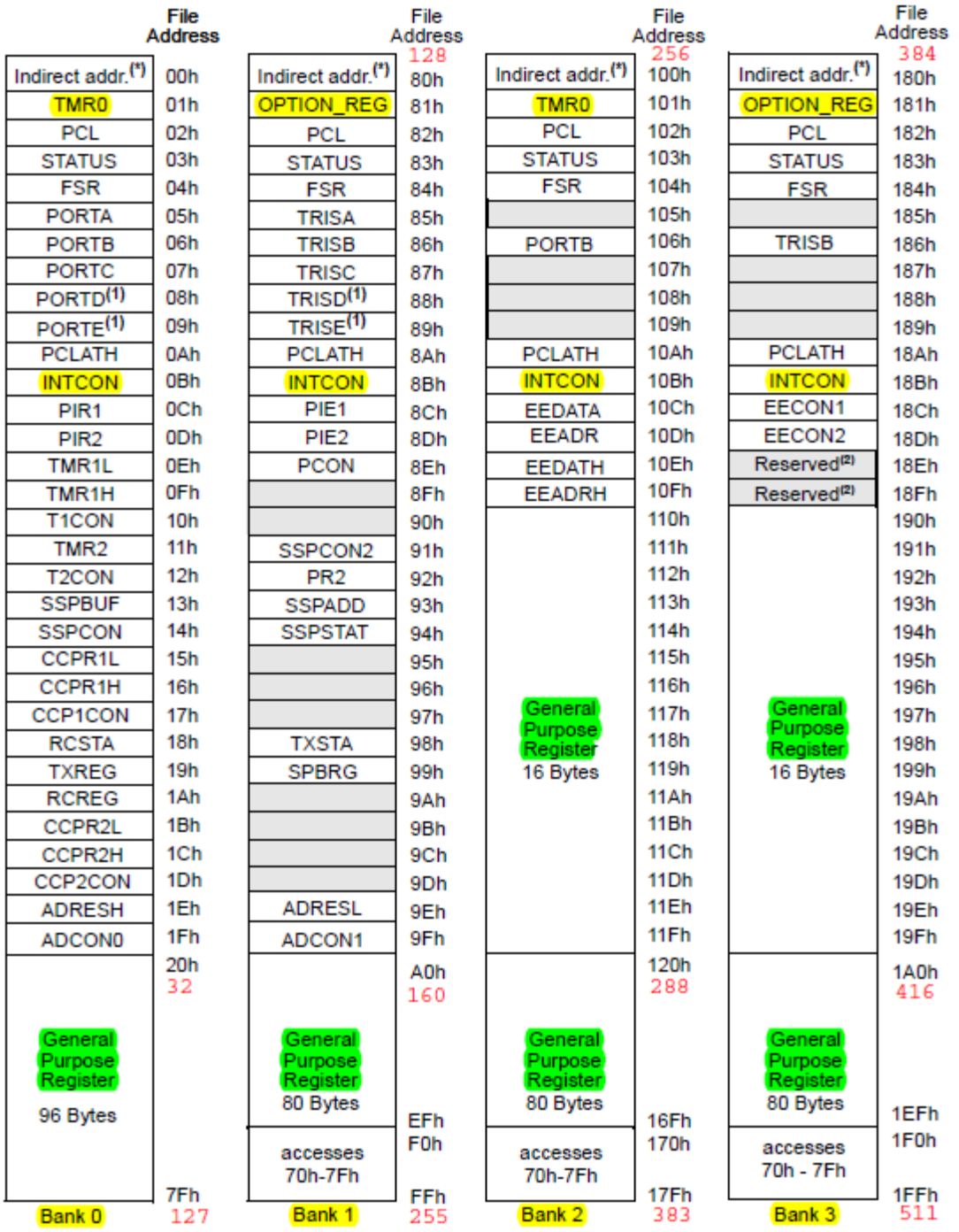
2. If *Prescale* = 256,

$$\text{Rollover Time} = 1.0851 \times 10^{-6} \times 256 \times 256 = 71.12\ \text{ms}$$

Timer0 Registers

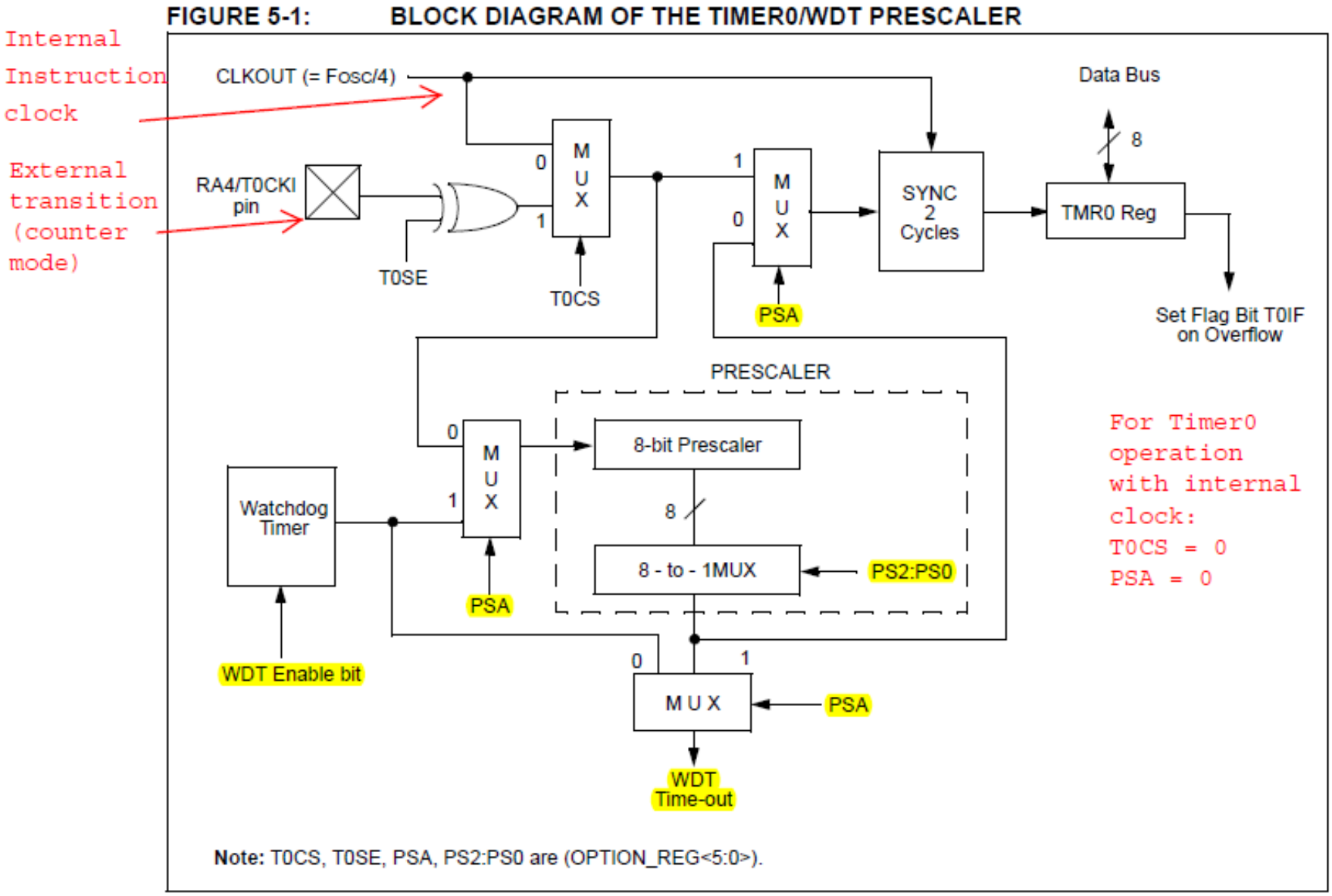
1. TMR0
2. OPTION_REG
3. INTCON

Data Memory



Data sheet, p. 13

Timer0 Block Diagram



Data sheet, p. 47

Timer0 Control Registers

REGISTER 5-1: OPTION_REG REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

bit 7 **$\overline{\text{RBPU}}$**

bit 6 **INTEDG**

bit 5 **T0CS**: TMR0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE**: TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA**: Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

Set *Prescale* to 256

```
movlw   B'10000111'

movwf   OPTION_REG
```

Datasheet, p. 19

Timer0 Control Registers

REGISTER 2-3: INTCON REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
bit 7							bit 0

Interrupt Control Register

bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

bit 6 **PEIE**: Peripheral Interrupt Enable bit
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts

bit 5 **T0IE**: TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

bit 4 **INTE**: RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt

bit 3 **RBIE**: RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2 **T0IF**: TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

Datasheet p. 20

Lab 2 Outline

1. Analog-to-digital conversion
2. Analog-to-digital conversion module
3. Timer 0
4. Data register bank selection
5. Lab 2 hardware setup

Bank Selection

1. The PIC16F877 has four data memory (RAM) banks.

2. The correct bank must be selected using the STATUS register in order to access a memory register.

REGISTER 2-1: STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)
1 = Bank 2, 3 (100h - 1FFh)
0 = Bank 0, 1 (00h - FFh)

bit 6-5 **RP1:RP0**: Register Bank Select bits (used for direct addressing)
11 = Bank 3 (180h - 1FFh)
10 = Bank 2 (100h - 17Fh)
01 = Bank 1 (80h - FFh)
00 = Bank 0 (00h - 7Fh)
Each bank is 128 bytes

RP: Register Pointer

Datasheet, p. 18

Bank Selection

REGISTER 2-1: STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)
1 = Bank 2, 3 (100h - 1FFh)
0 = Bank 0, 1 (00h - FFh)

bit 6-5 **RP1:RP0**: Register Bank Select bits (used for direct addressing)
11 = Bank 3 (180h - 1FFh)
10 = Bank 2 (100h - 17Fh)
01 = Bank 1 (80h - FFh)
00 = Bank 0 (00h - 7Fh)
Each bank is 128 bytes

Select Bank 1:

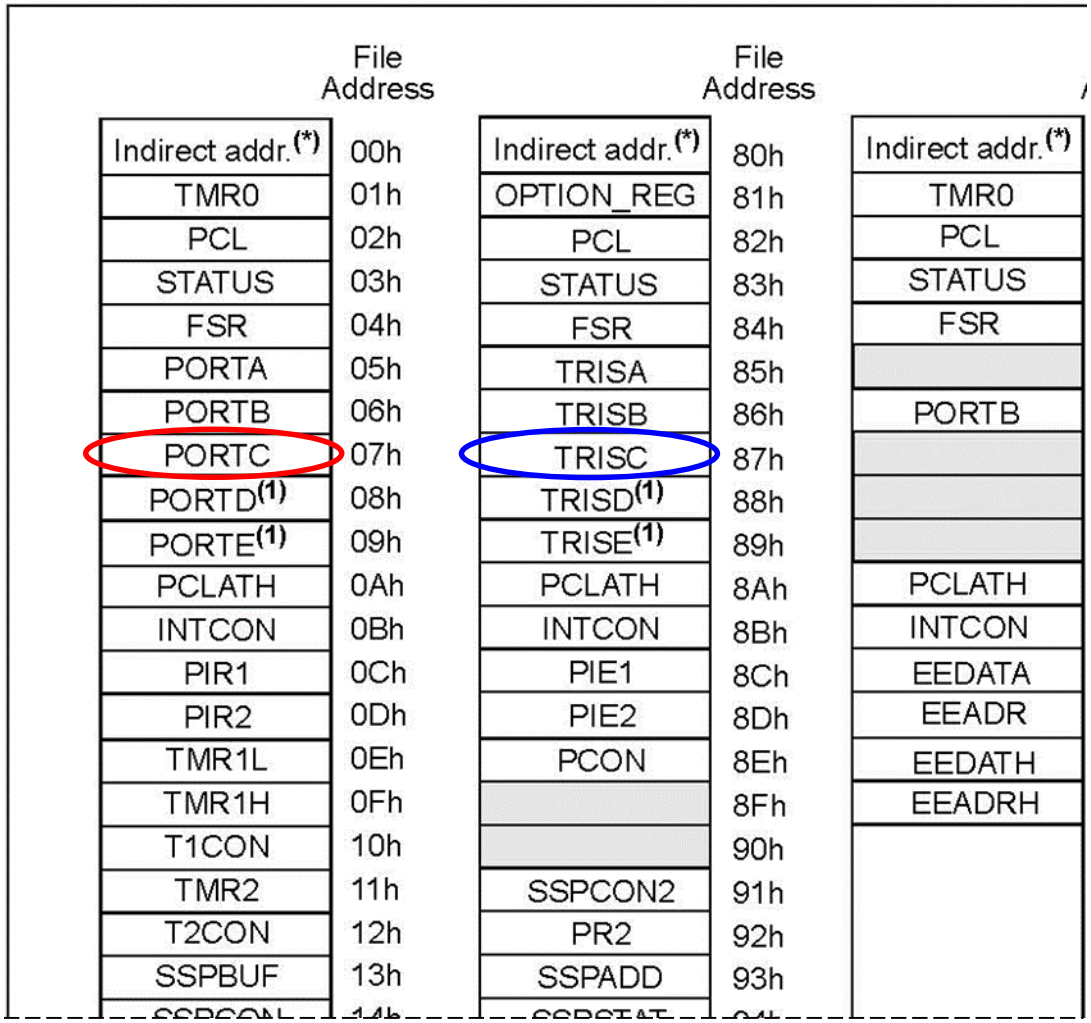
```
bcf     STATUS, RP1
bsf     STATUS, RP0
```

TABLE 13-2: PIC16F87X INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS						
ADDWF	f, d	Add W and f	1	00 0111 dfff ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00 0101 dfff ffff	Z	1,2
CLRF	f	Clear f	1	00 0001 lfff ffff	Z	2
CLRW	-	Clear W	1	00 0001 0xxx xxxx	Z	
COMF	f, d	Complement f	1	00 1001 dfff ffff	Z	1,2
DECF	f, d	Decrement f	1	00 0011 dfff ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00 1011 dfff ffff		1,2,3
INCF	f, d	Increment f	1	00 1010 dfff ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00 1111 dfff ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00 0100 dfff ffff	Z	1,2
MOVF	f, d	Move f	1	00 1000 dfff ffff	Z	1,2
MOVWF	f	Move W to f	1	00 0000 lfff ffff		
NOP	-	No Operation	1	00 0000 0xx0 0000		
RLF	f, d	Rotate Left f through Carry	1	00 1101 dfff ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00 1100 dfff ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00 0010 dfff ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00 1110 dfff ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00 0110 dfff ffff	Z	1,2

1. `movf f, d = 00 1000 dfff ffff`
2. Addresses in instructions are 7 bits (`fff ffff`)
3. But addresses in data memory are 9 bits ($2^9 = 512$)

FIGURE 2-3: PIC16F877/876 REGISTER FILE MAP



	File Address		File Address	
Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.(*)
TMR0	01h	OPTION_REG	81h	TMR0
PCL	02h	PCL	82h	PCL
STATUS	03h	STATUS	83h	STATUS
FSR	04h	FSR	84h	FSR
PORTA	05h	TRISA	85h	
PORTB	06h	TRISB	86h	PORTB
PORTC	07h	TRISC	87h	
PORTD(1)	08h	TRISD(1)	88h	
PORTE(1)	09h	TRISE(1)	89h	
PCLATH	0Ah	PCLATH	8Ah	PCLATH
INTCON	0Bh	INTCON	8Bh	INTCON
PIR1	0Ch	PIE1	8Ch	EEDATA
PIR2	0Dh	PIE2	8Dh	EEADR
TMR1L	0Eh	PCON	8Eh	EEDATH
TMR1H	0Fh		8Fh	EEADRH
T1CON	10h		90h	
TMR2	11h	SSPCON2	91h	
T2CON	12h	PR2	92h	
SSPBUF	13h	SSPADD	93h	

Bank 0 Bank 1

`TRISC` is at address 0x87 = 1000 0111

```
MOVF TRISC, W
00 1000 0fff ffff
```

7 bits for address →

```
00 1000 0000 0111
```

If Bank 0 is selected, this instruction results in the contents of `PORTC` being moved to W, not contents of `TRISC`.

Datasheet p. 13

FIGURE 2-3: PIC16F877/876 REGISTER FILE MAP

	File Address		File Address	
Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.
TMR0	01h	OPTION_REG	81h	TMR0
PCL	02h	PCL	82h	PCL
STATUS	03h	STATUS	83h	STATUS
FSR	04h	FSR	84h	FSR
PORTA	05h	TRISA	85h	
PORTB	06h	TRISB	86h	PORTB
PORTC	07h	TRISC	87h	
PORTD(1)	08h	TRISD(1)	88h	
PORTE(1)	09h	TRISE(1)	89h	
PCLATH	0Ah	PCLATH	8Ah	PCLATH
INTCON	0Bh	INTCON	8Bh	INTCON
PIR1	0Ch	PIE1	8Ch	EEDATA
PIR2	0Dh	PIE2	8Dh	EEADR
TMR1L	0Eh	PCON	8Eh	EEDATH
TMR1H	0Fh		8Fh	EEADRH
T1CON	10h		90h	
TMR2	11h	SSPCON2	91h	
T2CON	12h	PR2	92h	

Bank 1

How can we make the PIC move `TRISC` to W?

The instruction
00 1000 0000 0111 results in address 7, relative to current bank, being moved to `W`.

So if Bank 1 is selected, then `TRISC` will be moved to `W`.

The `banksel` Directive

1. **`banksel <register>`**

   - Determines which bank **`<register>`** is in.
   - Sets the **`STATUS<RP1:RP0>`** bits automatically

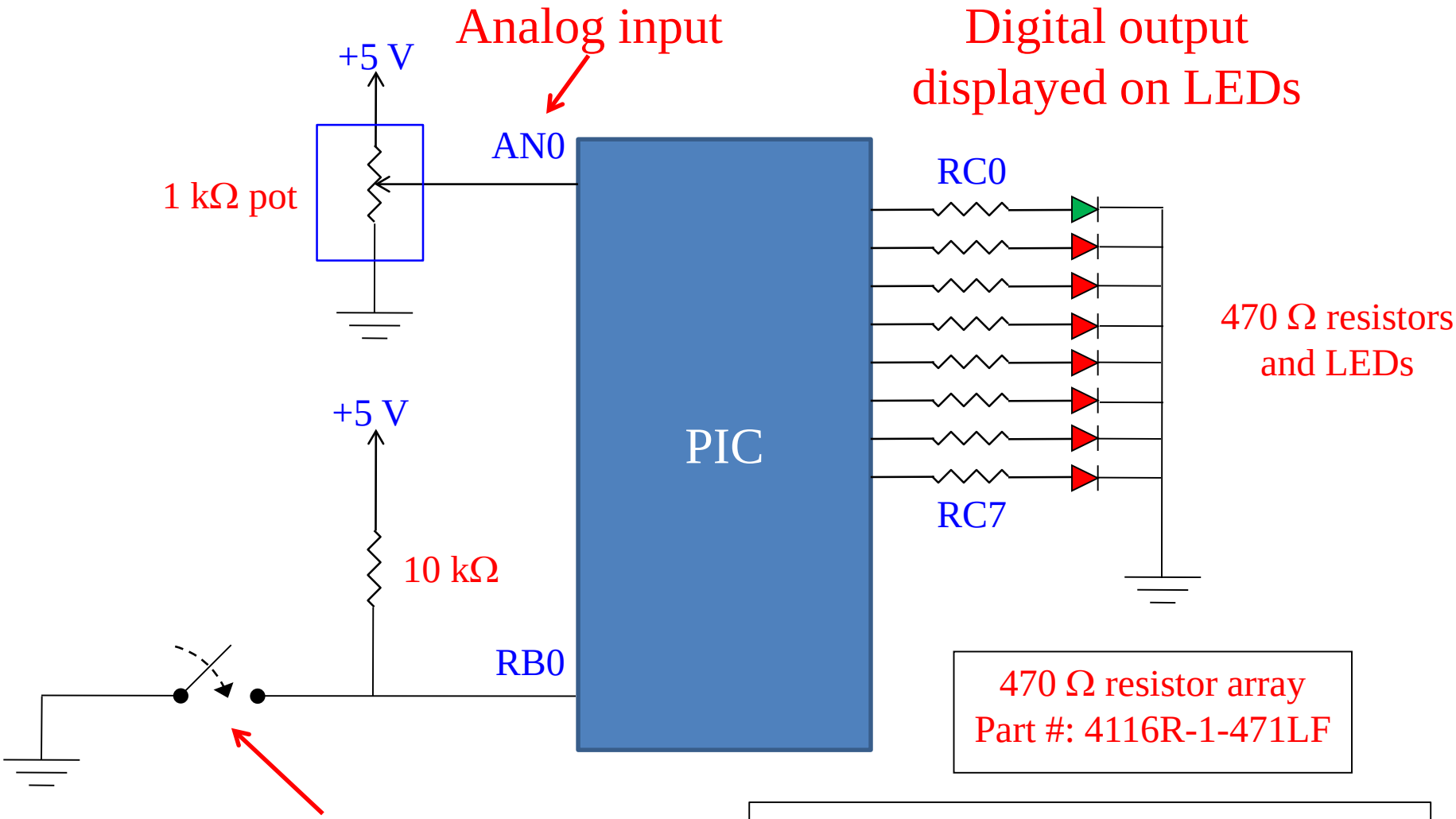
2. Example:

```
banksel     TRISC
movf        TRISC, W
```

Lab 2 Outline

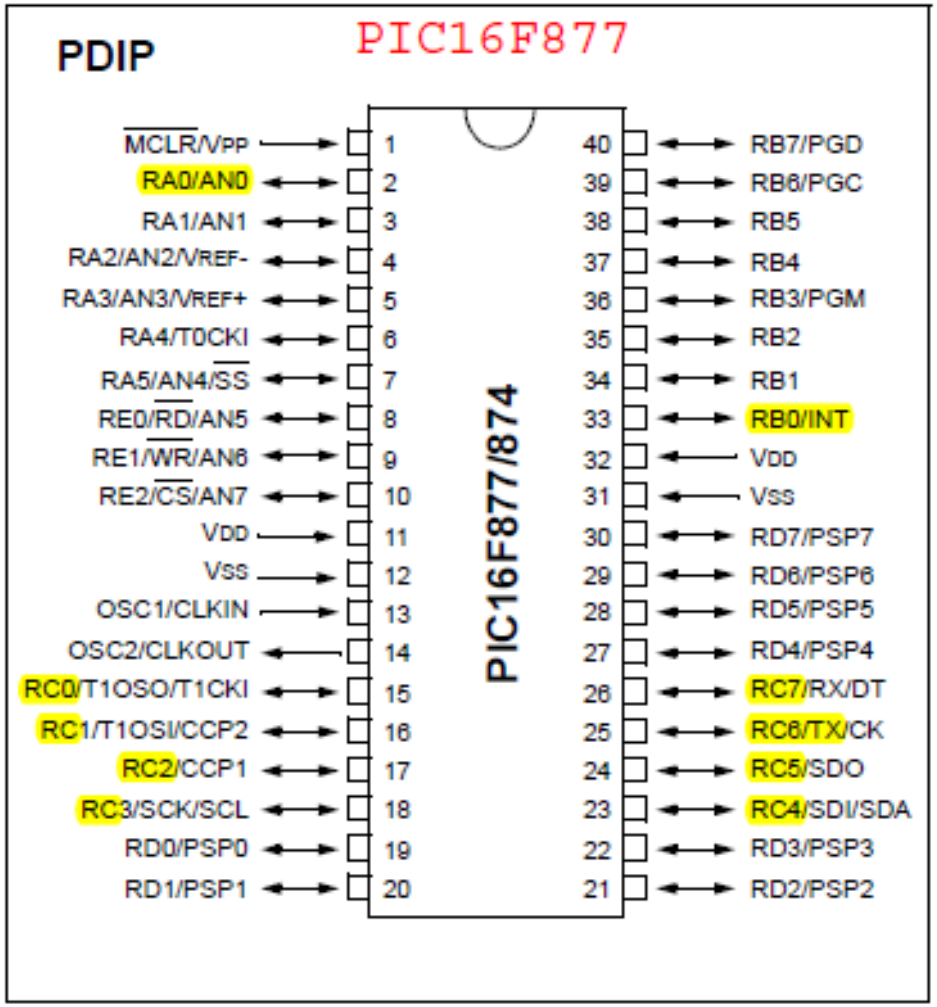
1. Analog-to-digital conversion
2. Analog-to-digital conversion module
3. Timer 0
4. Data register bank selection
5. Lab 2 hardware setup

Lab 2: Schematic

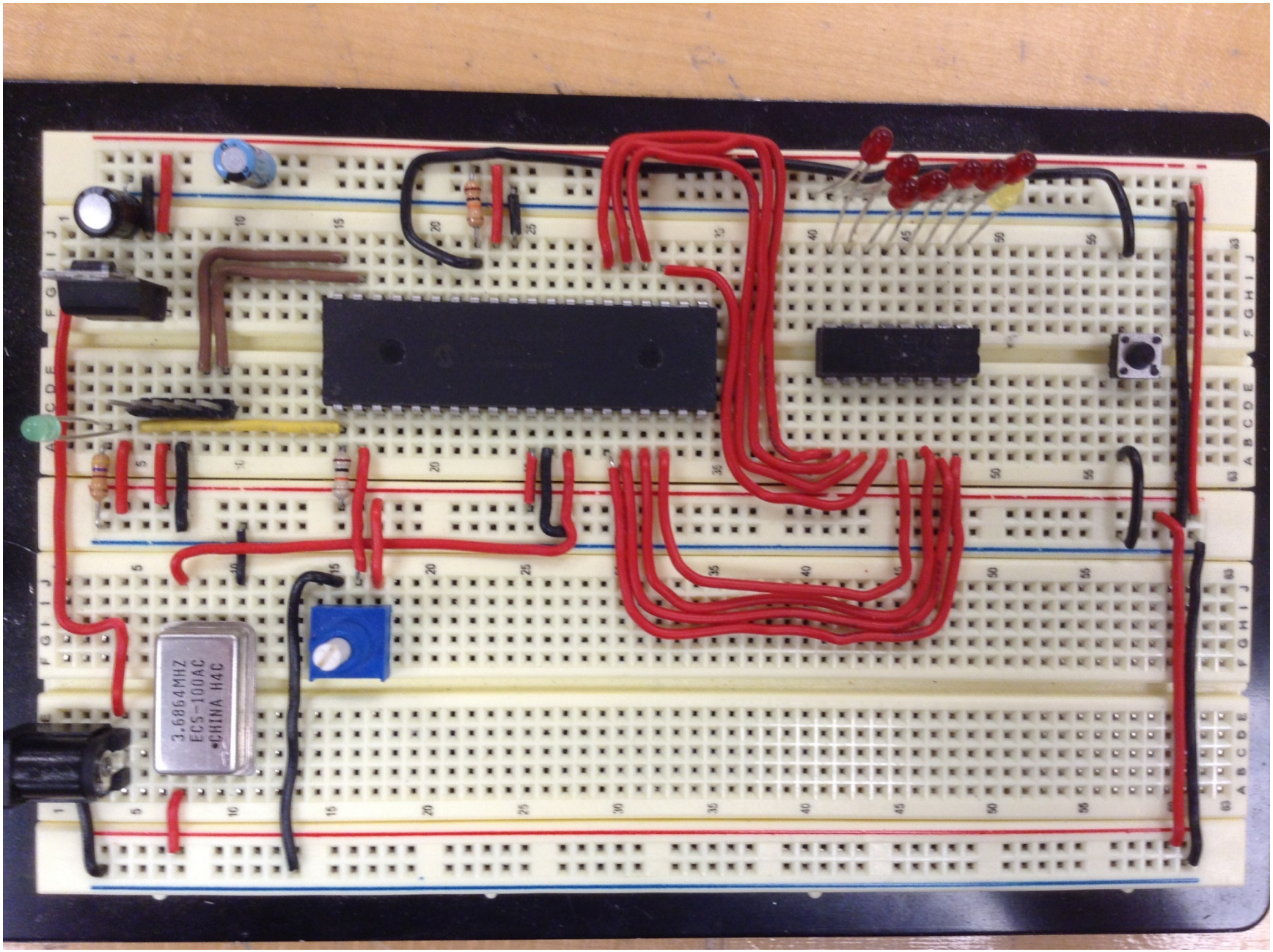


Lab 2 Pin Usage

Pin Diagram

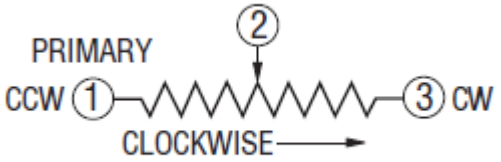


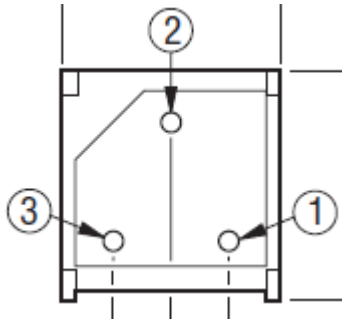
Lab 2 Hardware Layout



lab02.asm

1 KΩ potentiometer





Digi-Key Part Number
3310Y-001-102L-ND

```
Init
    banksel PORTC        ; Default --- Bank 0
    clrf    PORTC        ; Clear PORTC initially

    ; Set up the ADCON0 control register.

    movlw   B'01000001' ; Fosc/8, AN0, A/D enabled
    movwf   ADCON0       ; ADCON0 is in Bank 0

    ; Set up the Timer0 control register.

    banksel OPTION_REG   ; Switch to Bank 1
    movlw   B'10000111' ; Use the internal instruction clock,
                         ; prescaler is assigned to Timer0,
    movwf   OPTION_REG   ; prescaler: 1:256

    ; Set up the ADCON1 control register

    movlw   B'00001110' ; ADCON1 is in Bank 1.
                         ; Left justify <ADRESH:ADRESL>,
                         ; 1 analog channel (AN0), A/D on,
    movwf   ADCON1       ; use VDD and VSS references voltages.

    ; Set up PORTC for output to LEDs

    clrf    TRISC        ; TRISC is in Bank 1. All pins output.

    banksel PORTC        ; Return to default Bank 0
```

```
Main
    ; Timer0 delay for A/D voltage acquisition

    btfss   INTCON,T0IF ; Test the TIMER0 interrupt flag bit (T0IF).
                        ; The INTCON register is in Bank 0.
                        ; If T0IF = 1 (TMR0 rollover), skip next
                        ; instruction (in this case, the "goto"
                        ; instruction).
    goto    Main

    bcf     INTCON,T0IF ; Clear the T0IF bit for the next interrupt.

    banksel ADCON0      ; ADCON0 in Bank 0

    bsf     ADCON0,GO   ; Start the A/D conversion

WaitForConversion

    btfss   PIR1, ADIF  ; Test the A/D conversion-complete flag ADIF
                        ; in the PIR1 register. PIR1 is in Bank 0.
                        ; If ADIF = 1(conversion complete),
                        ; skip the "goto" instruction.

    goto    WaitForConversion

    bcf     PIR1, ADIF  ; Clear the ADIF bit for the next conversion.

    clrf    PORTC       ; Clear PORTC (Turn off the LEDs)

LoopWhilePushed         ; Loop if PORTB<0> = 0 (button pressed)

    btfss   PORTB, 0    ; If PORTB<0> = 1, skip the "goto" instruction.

    goto    LoopWhilePushed

    ; Otherwise, illuminate the LEDs according to ADRESH.
    ; Least-significant bits ignored.

    movf    ADRESH, W   ; Write A/D result to PORTC
    movwf   PORTC
    goto    Main        ; Repeat until HALT or Power Down

    end                 ; End of program
```

End of Lab 2