Chapter 8
Analog-to-Digital Converter
(ADC)

Why Analog-to-Digital Conversion?

- In the physical world, everything is analog (continuous).
- We need an analog-to-digital converter to translate analog signals to digital numbers so that the microcontroller can read and process them.

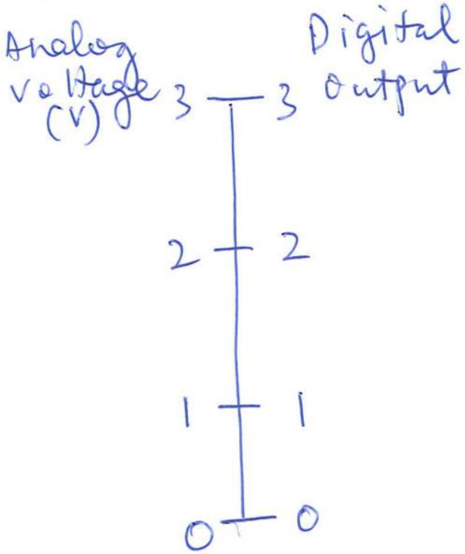
Example: 2-bit ADC

example 1

$V_{ref+} = 3V$

$V_{ref-} = 0V$

$n = 2$



$$V_k = \left(\frac{3-0}{3}\right) k = k$$

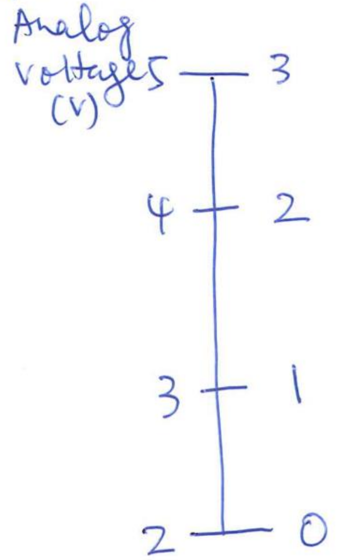
V_k = voltage level corresponding to the digital representation k

example 2

$V_{ref+} = 5V$

$V_{ref-} = 2V$

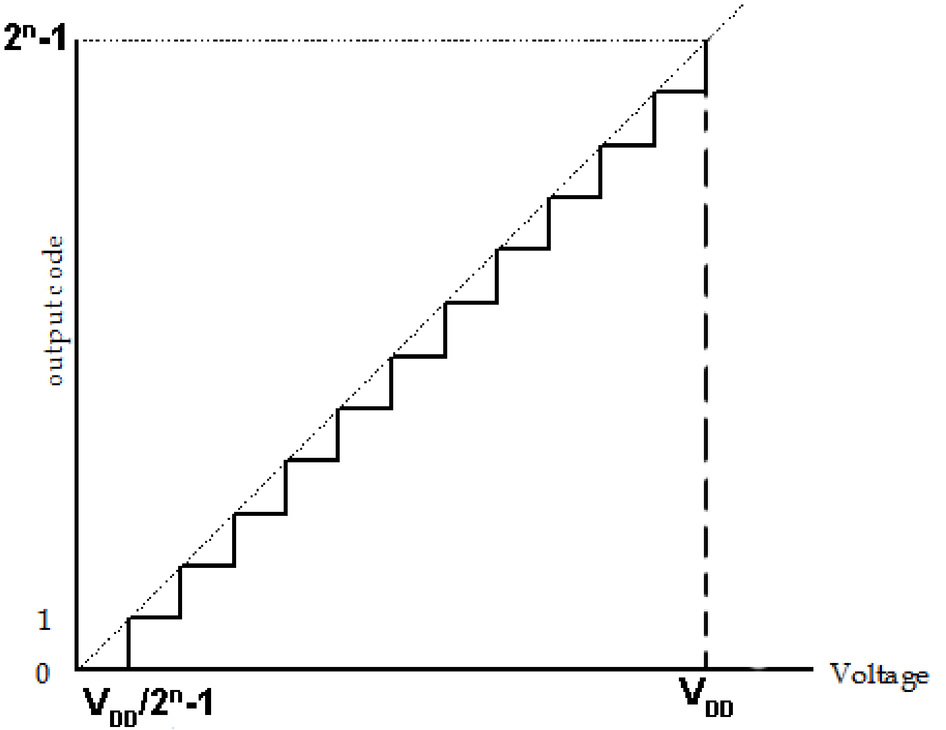
$n = 2$



$$V_k = 2 + \left(\frac{3-0}{3}\right) k$$
$$= 2 + k$$

n-resolution ADC

	Analog voltage (V)	Digital represen-tation
V_{REF-}	0	0
V_{REF+}	V_{DD}	2^n-1
V_k	$k\left(\frac{V_{DD}}{2^n - 1}\right)$	k



Or in terms of V_{REF-} & V_{REF+}:

$$V_k = V_{REF-} + k\left(\frac{V_{REF+} - V_{REF-}}{2^n - 1}\right)$$

Examples

e.g., $V_{REF-} = 0$, $V_{REF+} = 5V$, 10-bit resolution ADC. Suppose the ADC results are as follows. What are the corresponding voltages?

a. 20

$$V_{20} = 20x5/(2^{10}-1) = 0.0977V$$

b. 499

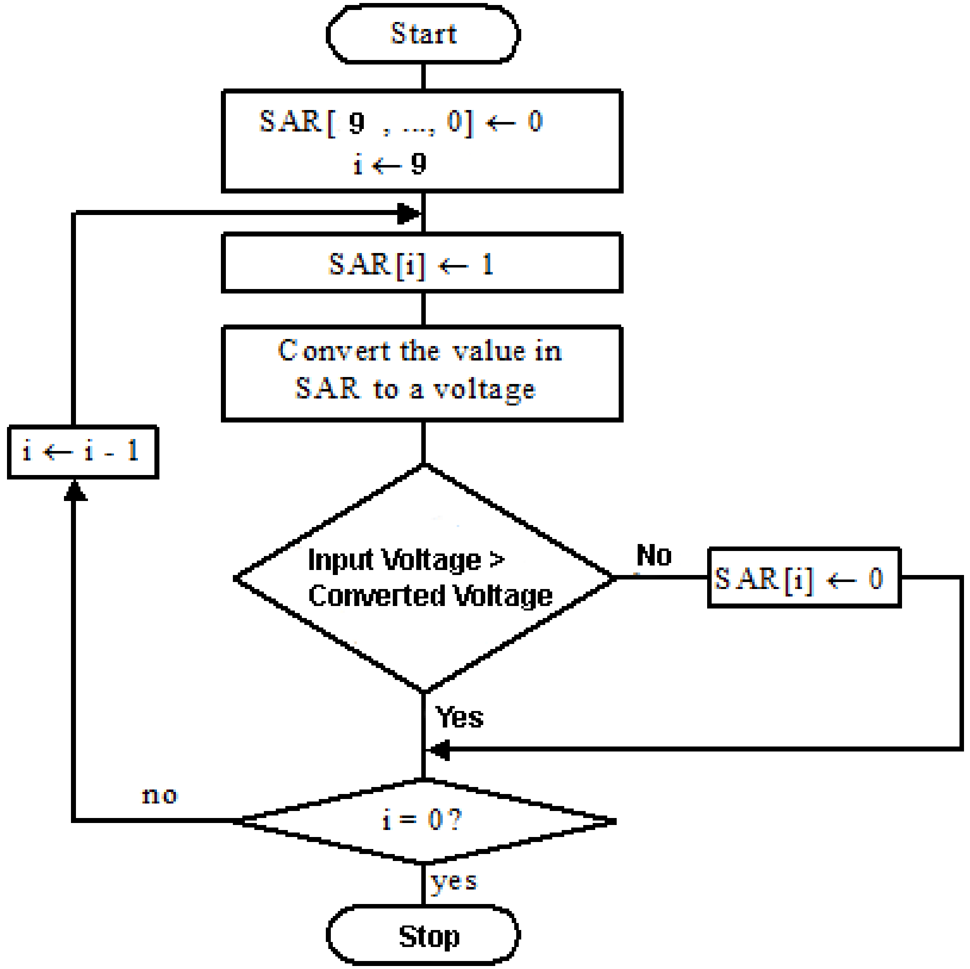
$$V_{499} = 499x5/(2^{10}-1)= 2.43V$$

c. 898

$$V_{898} = 898x5/(2^{10}-1)=4.38V$$

How can ADC be done?

Successive Approximation



Examples of Successive Approximation

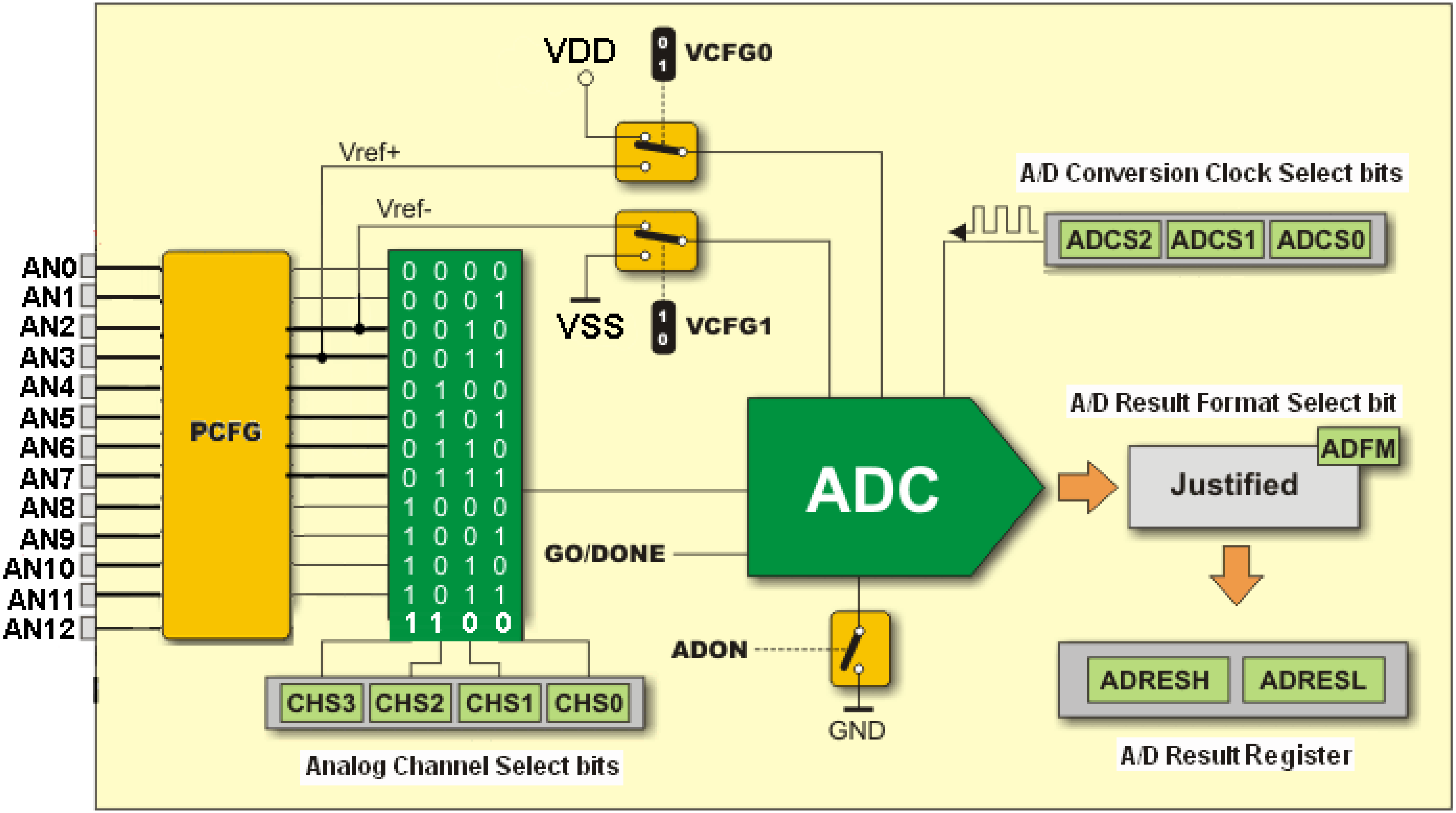
- 2 bit representation → 4 Levels
- $V_{REF-} = V_0$ → 0 V and $V_{REF+} = V_3$ → 3 V
- Example 1: Input Voltage 2.3V
  - 1st iteration: SAR = 10 → 2.3 V > 2 V? Yes
  - 2nd iteration: SAR = 11 → 2.3 > 3 V? No → SAR[0] = 0 → Final SAR = 10
- Example 2: Input Voltage 1.7 V
  - 1st iteration: SAR = 10 → 1.7 > 2 V? No → SAR[1] = 0 (i.e., SAR = 00)
  - 2nd iteration: SAR = 01 → 1.7 > 1 V? Yes → Final SAR = 01.

PIC18 ADC Module

- PIC18F4520 has 13 analog inputs channels. (Only one channel is used at any given time)
- Generates 10-bit binary results
- V_{REF-} & V_{REF+} are adjustable → Minimum resolution can be designed.
- Note: Minimum resolution

$$\frac{V_{REF+} - V_{REF-}}{2^n - 1}$$

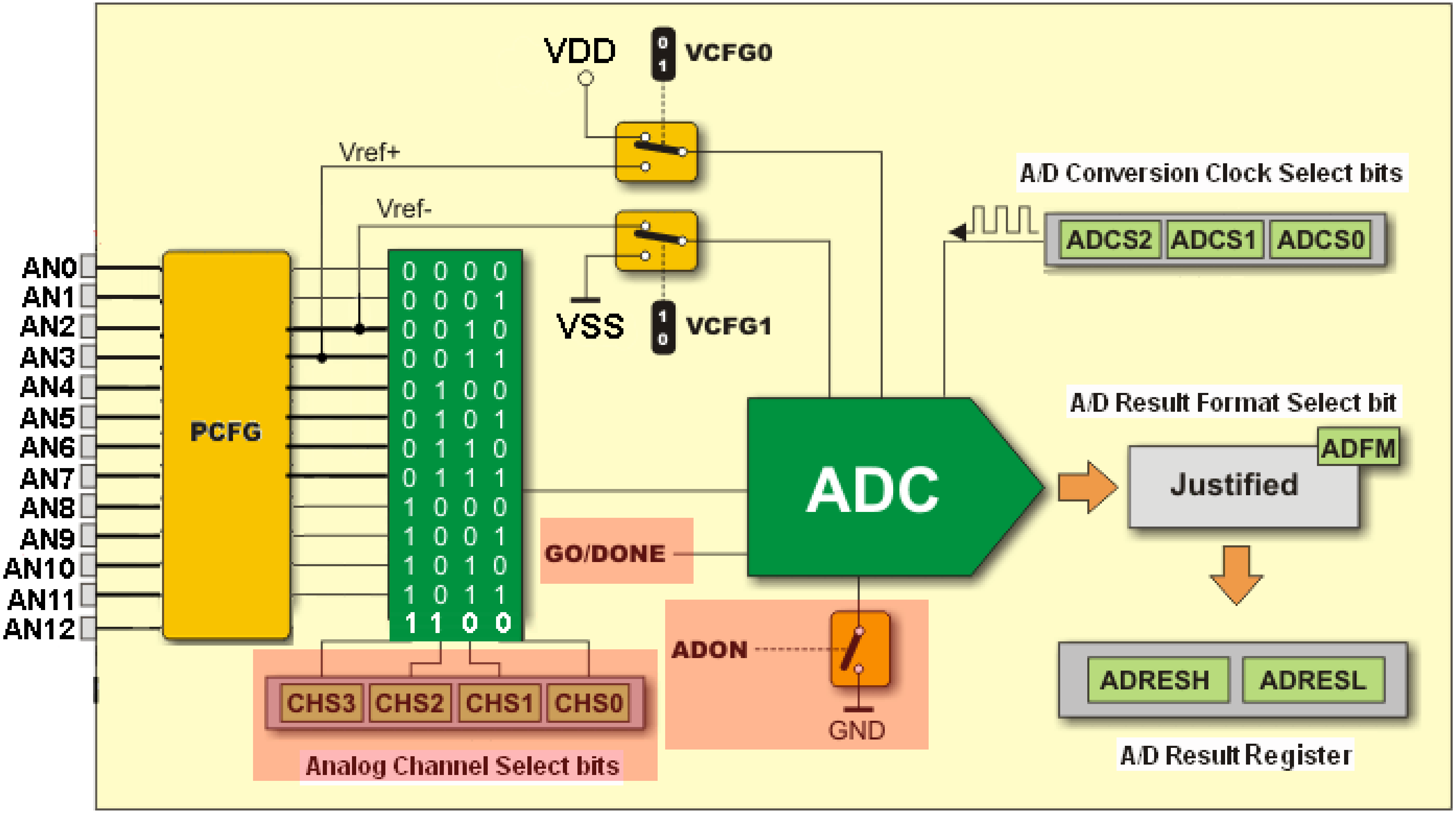
Block Diagram of PIC18 ADC



SFR involving in ADC

- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)
- A/D Control Register 2 (ADCON2)
- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)

ADCON0



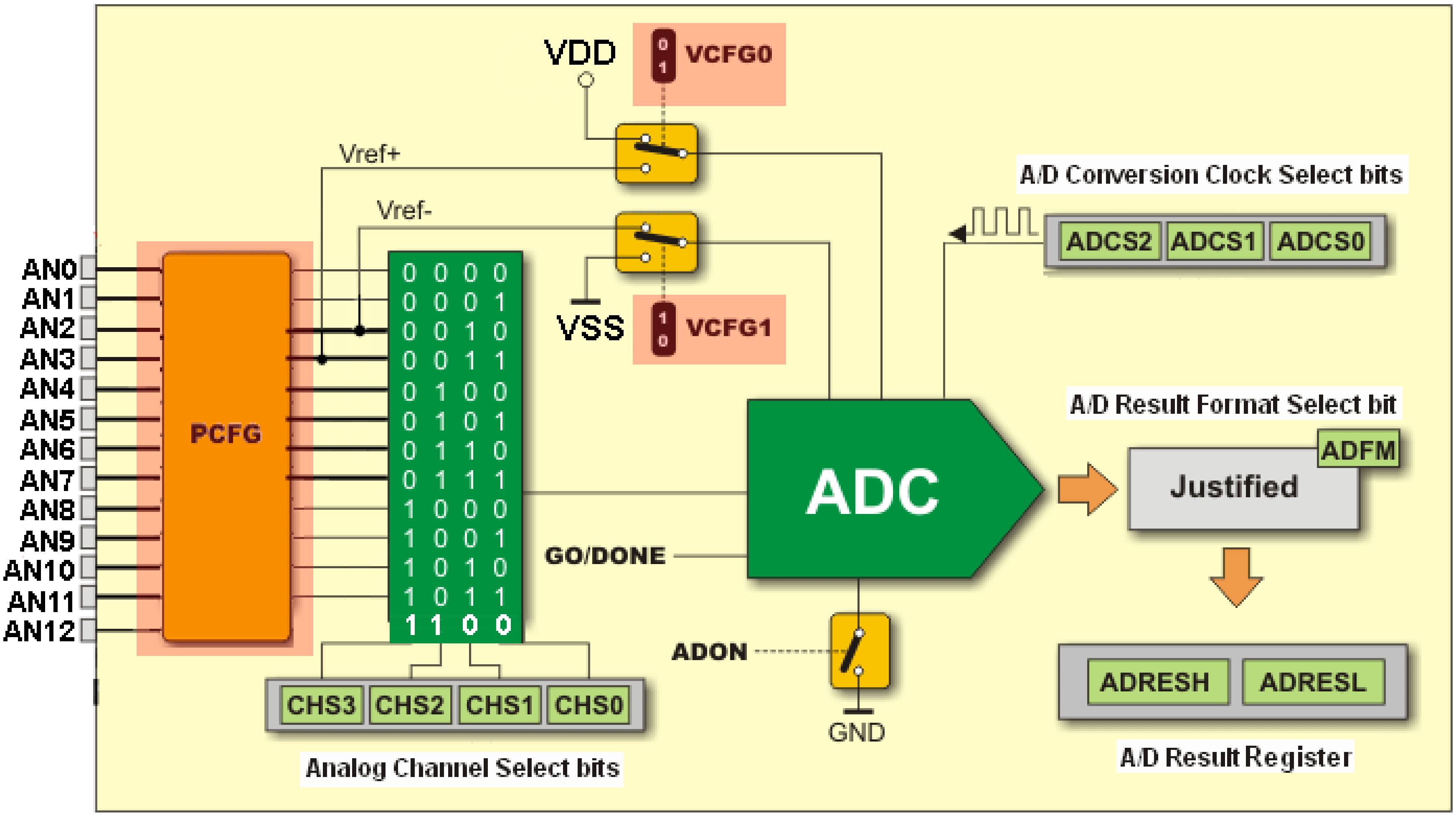
VDD
VCFG0
Vref+
Vref-
VSS
VCFG1
A/D Conversion Clock Select bits
ADCS2
ADCS1
ADCS0
AN0
AN1
AN2
AN3
AN4
AN5
AN6
AN7
AN8
AN9
AN10
AN11
AN12
PCFG
0 0 0 0
0 0 0 1
0 0 1 0
0 0 1 1
0 1 0 0
0 1 0 1
0 1 1 0
0 1 1 1
1 0 0 0
1 0 0 1
1 0 1 0
1 0 1 1
1 1 0 0
ADC
A/D Result Format Select bit
ADFM
Justified
GO/DONE
ADON
GND
CHS3
CHS2
CHS1
CHS0
Analog Channel Select bits
ADRESH
ADRESL
A/D Result Register

ADCON0

			R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	Features
ADCON0	-	-	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	**Bit name**
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

- CHS3:CHS0: Analog Channel Select Bits
  - PIC18F4520 has only 1 Analog-to-Digital unit
  - Can only select 1 input from AN0:AN12 to be sampled and converted
- GO/DONE:
  - Setting this bit starts A/D conversion
  - Cleared when A/D conversion is done
- ADON:
  - Power up (1)/Shut off (0) the ADC module

ADCON1



VDD
VCFG0
Vref+
Vref-
A/D Conversion Clock Select bits
ADCS2
ADCS1
ADCS0
VSS
VCFG1
AN0
AN1
AN2
AN3
AN4
AN5
AN6
AN7
AN8
AN9
AN10
AN11
AN12
PCFG
0 0 0 0
0 0 0 1
0 0 1 0
0 0 1 1
0 1 0 0
0 1 0 1
0 1 1 0
0 1 1 1
1 0 0 0
1 0 0 1
1 0 1 0
1 0 1 1
1 1 0 0
ADC
A/D Result Format Select bit
ADFM
Justified
GO/DONE
ADON
GND
CHS3
CHS2
CHS1
CHS0
Analog Channel Select bits
ADRESH
ADRESL
A/D Result Register

ADCON1

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
			R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	Features
ADCON1	-	-	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	Bit name

<u>Voltage Reference Configuration bits</u>

- VCFG1: V_{REF-} source

  1 = V_{REF-} (AN2)

  0 = V_{SS}

- VCFG0: V_{REF+} source

  1 = V_{REF+} (AN3)

  0 = V_{DD}

<u>A/D Port Configuration bits</u>

PCFG<3:0> – The input analog signal must be configured as an analog input.

ADCON1: A/D Port Configuration bits

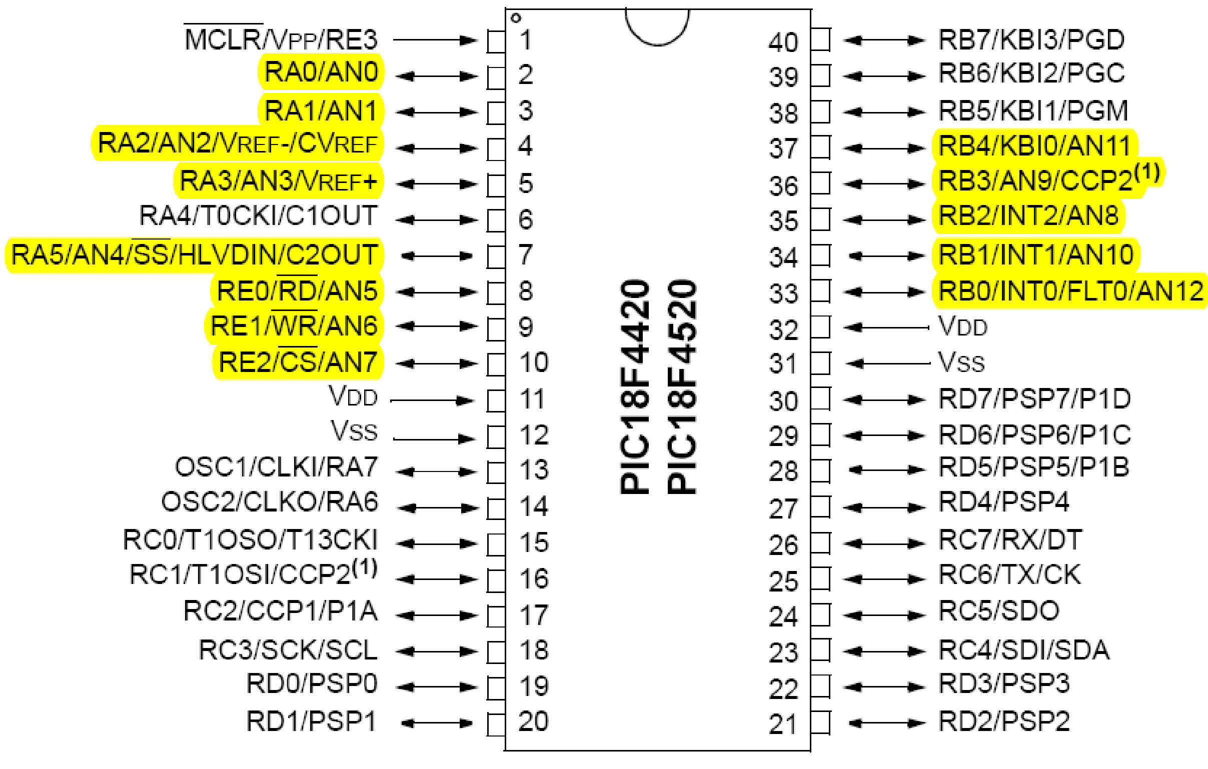
PCFG3: PCFG0	AN12	AN11	AN10	AN9	AN8	AN7(2)	AN6(2)	AN5(2)	AN4	AN3	AN2	AN1	AN0
0000(1)	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111(1)	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	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	A	A
1110	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

A = Analog input D = Digital I/O

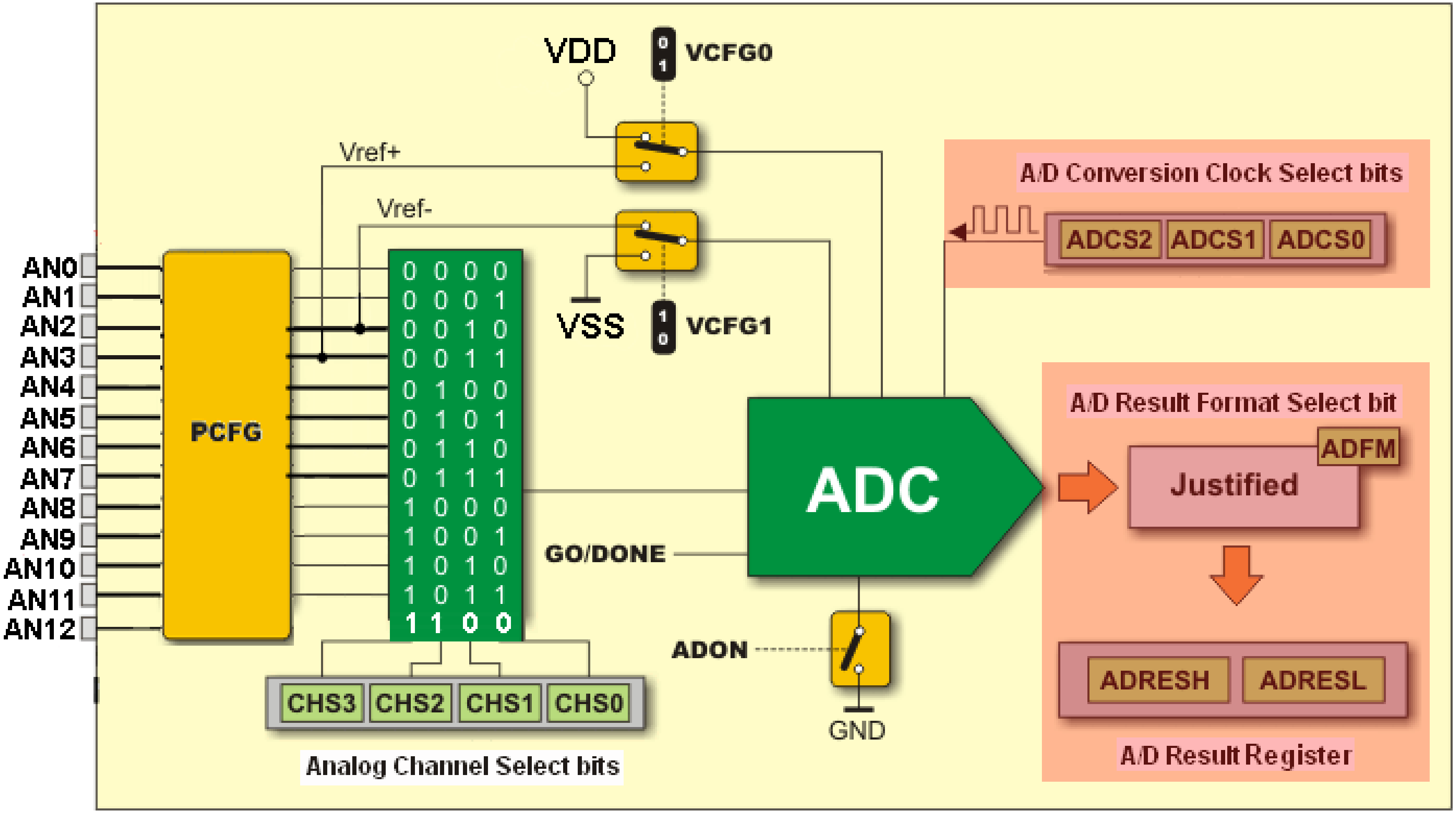
ADCON1: A/D Port Configuration bits

- Do the following two lines sound familiar?

```
movlw 0x0F
movwf ADCON1
```



ADCON2

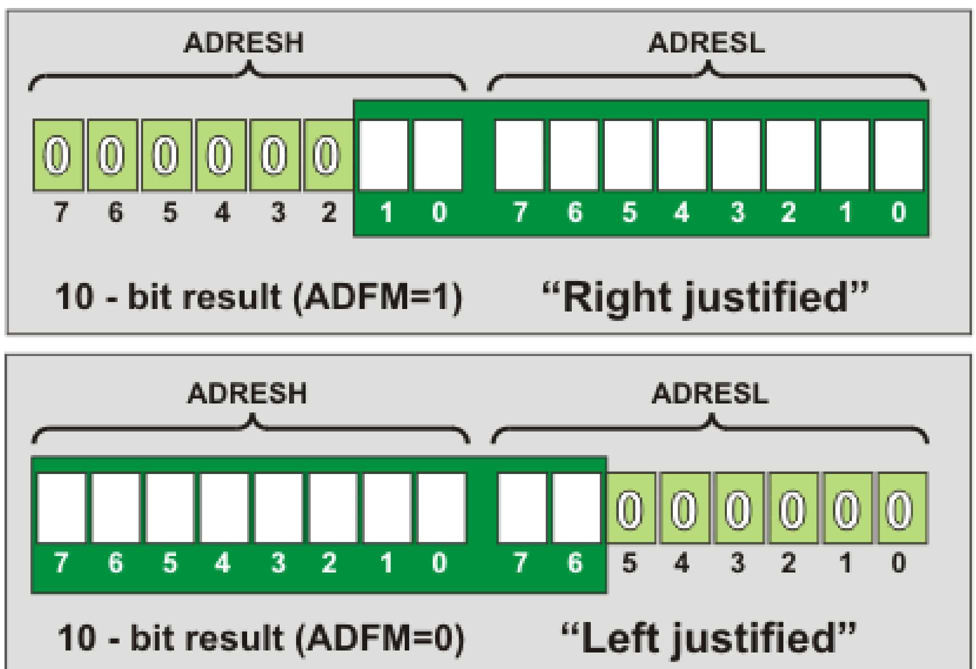


VDD
VCFG0
Vref+
Vref-
VSS
VCFG1
A/D Conversion Clock Select bits
ADCS2
ADCS1
ADCS0
AN0
AN1
AN2
AN3
AN4
AN5
AN6
AN7
AN8
AN9
AN10
AN11
AN12
PCFG
0 0 0 0
0 0 0 1
0 0 1 0
0 0 1 1
0 1 0 0
0 1 0 1
0 1 1 0
0 1 1 1
1 0 0 0
1 0 0 1
1 0 1 0
1 0 1 1
1 1 0 0
ADC
GO/DONE
A/D Result Format Select bit
ADFM
Justified
ADON
GND
CHS3
CHS2
CHS1
CHS0
Analog Channel Select bits
ADRESH
ADRESL
A/D Result Register

ADCON2

	R/W (0)		R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	Features
ADCON2	ADFM	-	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	**Bit name**
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

- ADFM: A/D Result Format Select bit



A/D Conversion Time: T_{AD}

- T_{AD} = time required to complete 1-bit A/D conversion
- Time required for one full 10-bit A/D conversion is 11 to 12 T_{AD} – <u>Successive approximation</u>
- From the PIC18F4520 specification, T_{AD} must be between 0.7 – 25 μs.
- T_{AD} can be specified from $2T_{osc}$ to $64T_{osc}$ by selecting the *A/D Conversion Clock Select Bits (*ADCS<0:2> in ADCON2*).*

How should T_{AD} be set?

- e.g., If a 4MHz clock is installed, T_{osc} = 0.25μs.

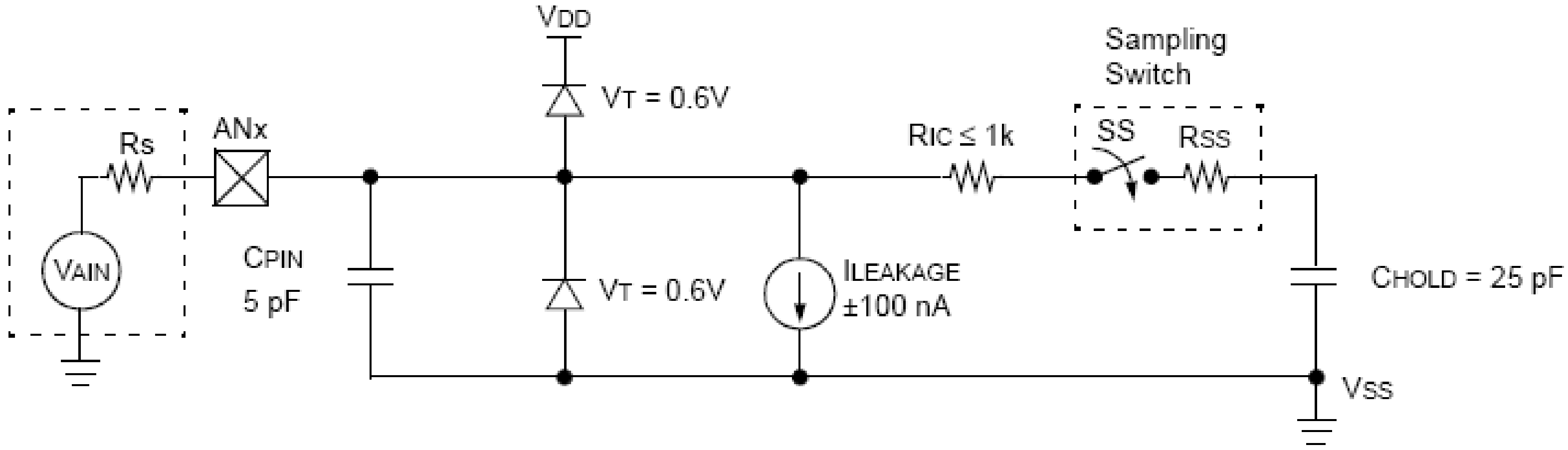
$$T_{AD} = kT_{osc} \geq 0.7\mu s$$

$$k \geq 0.7/0.25 = 2.8$$

- The minimum T_{AD} that can be set is $4T_{osc}$.
- T_{AD} can be chosen from $4T_{osc}$ (1μs) to $64T_{osc}$ (16μs).

Sample-and-hold circuit

- The analog input voltage will charge up C_{HOLD} to a steady state before ADC begins.
- The time required to reach steady state depends on R_s, R_{IC}, R_{ss} and C_{HOLD}.



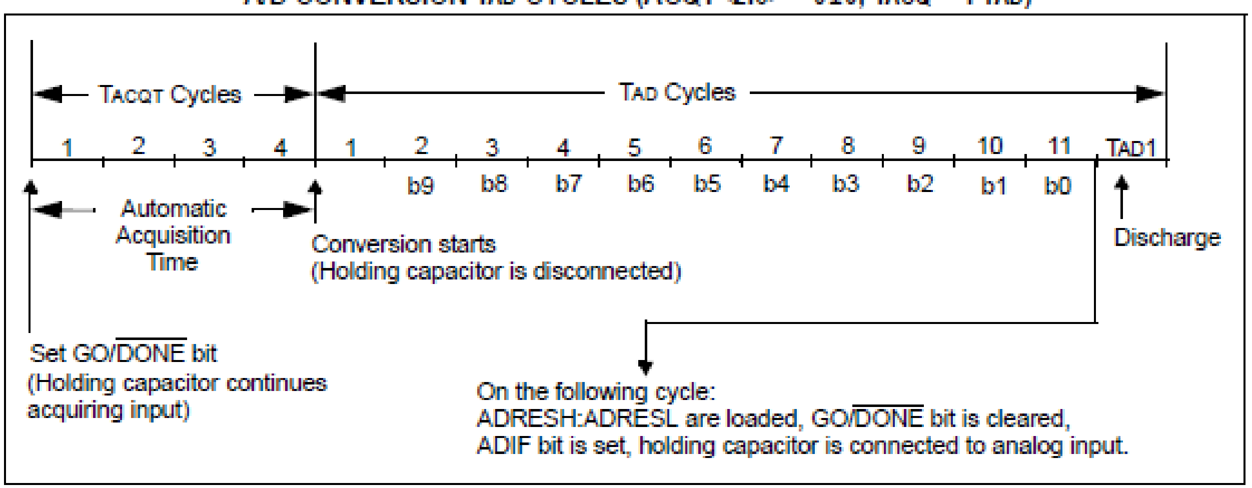
A/D Acquisition Time (ACQT)

- The minimum acquisition time (T_{ACQ}) for PIC18F4520 is 1.4µs.
- ADC module will insert an acquisition time between the GO/DONE bit is set and when the conversion starts.
- Suppose T_{AD} = 1µs. T_{ACQ} >= 2 T_{AD}
- T_{ACQ} is set by setting ACQT<0:2> of ADCON2 (chosen as a multiple of T_{AD}).

A/D Conversion Process

1. Set GO/DONE bit
2. Wait T_{ACQ} to charge up C_{HOLD}
3. C_{HOLD} is disconnected and ADC begins
4. 11-12 T_{AD} is required for a 10-bit ADC
5. When finished:
   - ADRESH:ADRESL are loaded
   - GO/DONE bit is cleared
   - ADIF bit is set
   - C_{HOLD} is reconnected to the analog input

A/D CONVERSION TAD CYCLES (ACQT<2:0> = 010, TACQ = 4 TAD)



Example Use of the ADC

```
Main:     movlw   b'00001110'        ;Set RA0 Analog Port, others Digital I/O
          movwf   ADCON1
          movlw   b'00000001'        ;Select ADC Channel 0, Enable ADC
          movwf   ADCON0
          movlw   b'00010100'        ; ADFM Left justified, ACQT 4TAD,
          movwf   ADCON2             FOSC/4

          clrf TRISD                 ; set PORTD output

MainLoop: bsf ADCON0, GO             ; start Conversion
adc_wait: btfsc   ADCON0, GO         ; adc_wait;wait ADC to be done
            bra   adc_wait

          movff ADRESH, PORTD        ;display Top 8 bit

          bra MainLoop
```

Example

Assume f_{OSC} = 4MHz, write a few instructions to turn on and configure a PIC ADC with the following requirements:

- ADC converts the analog signal fed into Channel AN3.
- ADC result is left justified.
- Select V_{DD} and V_{SS} as reference voltages.
- T_{AD} must be at least 2.5 μs.
- T_{ACQ} must be at least 18 μs.

Example

$$ADCON0 = 00\underbrace{0011}_{\text{Channel 3}}01$$

$$ADCON1 = 00\underbrace{0}_{V_{SS}}\underbrace{0}_{V_{DD}}\underbrace{1011}_{\text{AN3 Digital}}$$

$$T_{AD} = k\,T_{OSC} = k(0.25\,\mu s) \geq 2.5\,\mu s$$

$$k \geq 10$$

choose $k = 16$

$$\boxed{T_{AD} = 4\,\mu s}$$

$$T_{ACQ} = k\,T_{AD} = k(4\,\mu s) \geq 18\,\mu s$$

$$k \geq 4.5$$

Choose $k = 6$

$$\boxed{T_{AD} = 24\,\mu s}$$

$$ADCON2 = \underbrace{0}_{\text{left justified}}\,0\,\underbrace{011}_{T_{ACQ}=6T_{AD}}\,\underbrace{101}_{T_{AD}=16T_{OSC}}$$

$$F_{AD} = \frac{F_{OSC}}{16}$$

You should be able to ...

- Describe the roles of the following 5 registers: ADCON0, ADCON1, ADCON2, ADRESH and ADRESL in the A/D conversion. These registers specify the following options:
  - Which channel to convert?
  - Which channel should be set to accept analog input?
  - How to store the conversion result?
  - What is V_{ref+} and V_{ref-}?
  - What is the conversion time (T_{AD})?
  - What is the acquisition time (T_{ACQ})?

You should be able to ...

- Explain the process of sampling analog voltage before A/D conversion.
- Explain why an acquisition time is needed before A/D conversion.
- Describe the successive approximation algorithm.
- Describe the whole A/D conversion process.