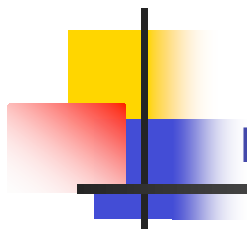




# Analog to Digital Converters

---

By: Dapeng Zhu  
Todd Sifleet  
Travis Nunnally  
Yachun Huang



# Introduction of ADC

By: Dapeng Zhu

---

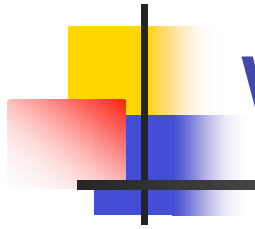
- What is ADC(Analog to Digital Converter)
- Why ADC is needed
- Application of ADC
- A/D conversion process



# What is ADC

---

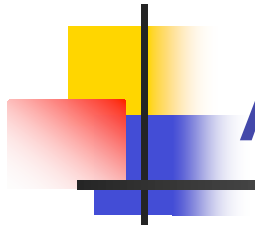
- An electronic integrated circuit which transforms a signal from analog (continuous) to digital (discrete) form.
- Analog signals are directly measurable quantities.
- Digital signals only have two states. For digital computer, we refer to binary states, 0 and 1.



# Why ADC is needed

---

- Microprocessors can only perform complex processing on digitized signals.
- When signals are in digital form they are less susceptible to the deleterious effects of additive noise.
- ADC Provides a link between the analog world of transducers and the digital world of signal processing and data handling.

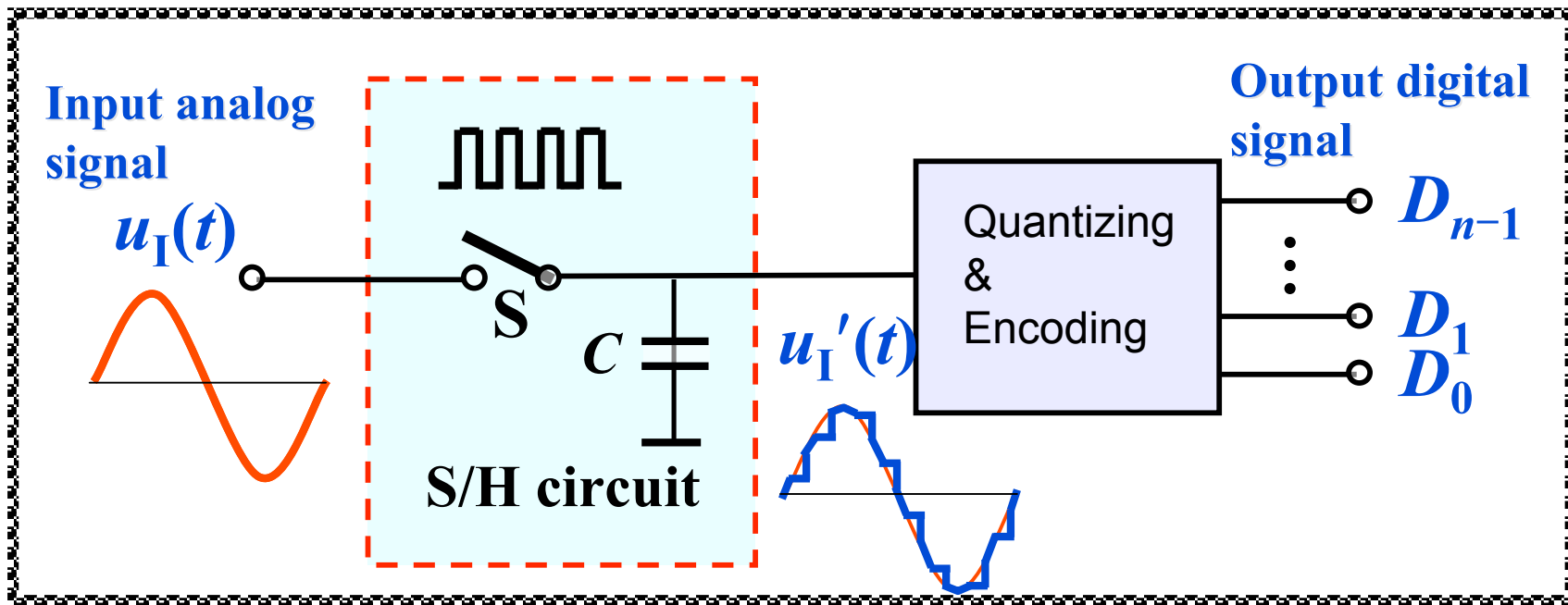


# Application of ADC

---

- ADC are used virtually everywhere where an analog signal has to be processed, stored, or transported in digital form.
- Some examples of ADC usage are digital volt meters, cell phone, thermocouples, and digital oscilloscope.
- Microcontrollers commonly use 8, 10, 12, or 16 bit ADCs, our micro controller uses an 8 or 10 bit ADC.

# ADC process

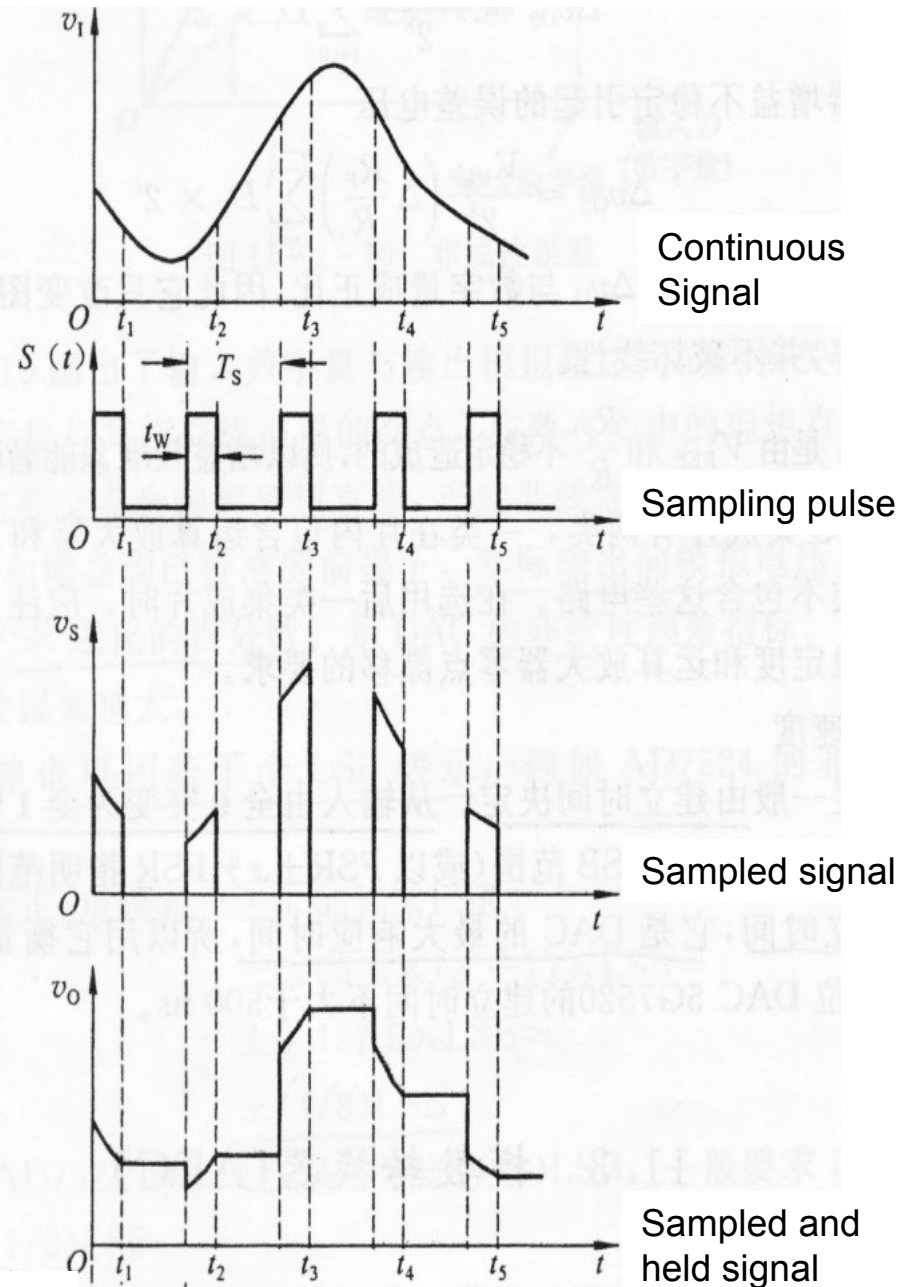


2 steps

- Sampling and Holding (S/H)
- Quantizing and Encoding (Q/E)

# Sampling and Holding

- Holding signal benefits the accuracy of the A/D conversion
- Minimum sampling rate should be at least twice the highest data frequency of the analog signal





# Quantizing and Encoding

---

Resolution:

The smallest change in analog signal that will result in a change in the digital output.

$$\Delta V = \frac{V_r}{2^N}$$

$V$  = Reference voltage range

$N$  = Number of bits in digital output.

$2^N$  = Number of states.

$\Delta V$  = Resolution

The resolution represents the quantization error inherent in the conversion of the signal to digital form



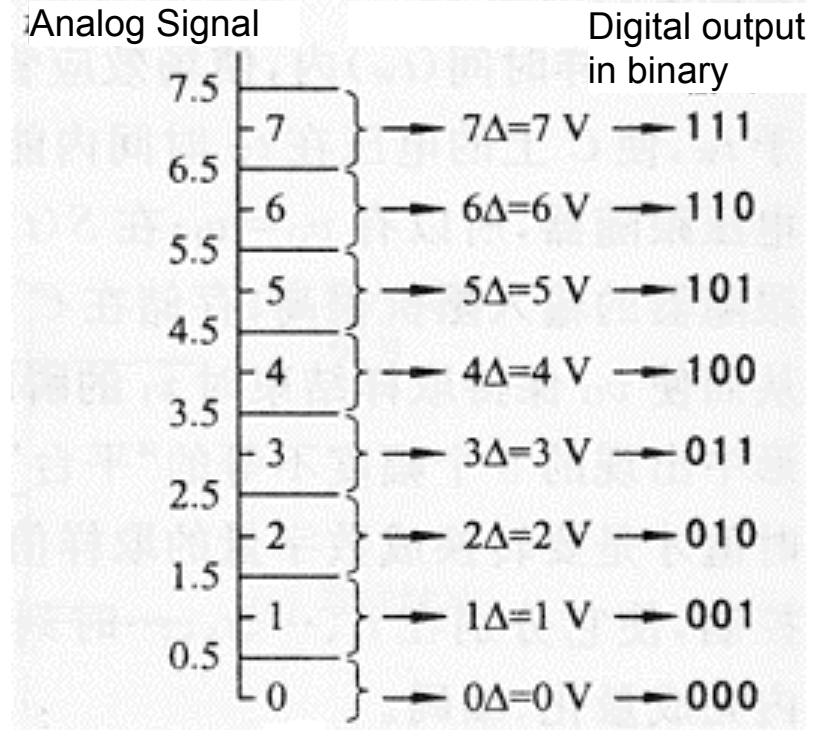
# Quantizing and Encoding

- Quantizing:

Partitioning the reference signal range into a number of discrete quanta, then matching the input signal to the correct quantum.

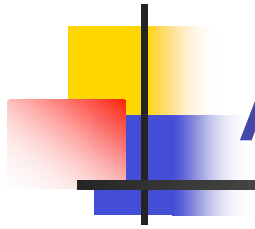
- Encoding:

Assigning a unique digital code to each quantum, then allocating the digital code to the input signal.



$$\Delta V = 1\text{ V}$$

$$\text{Maximum Quantization error} = \pm \frac{1}{2} \Delta V = \pm 0.5\text{ V}$$



# Accuracy of A/D Conversion

---

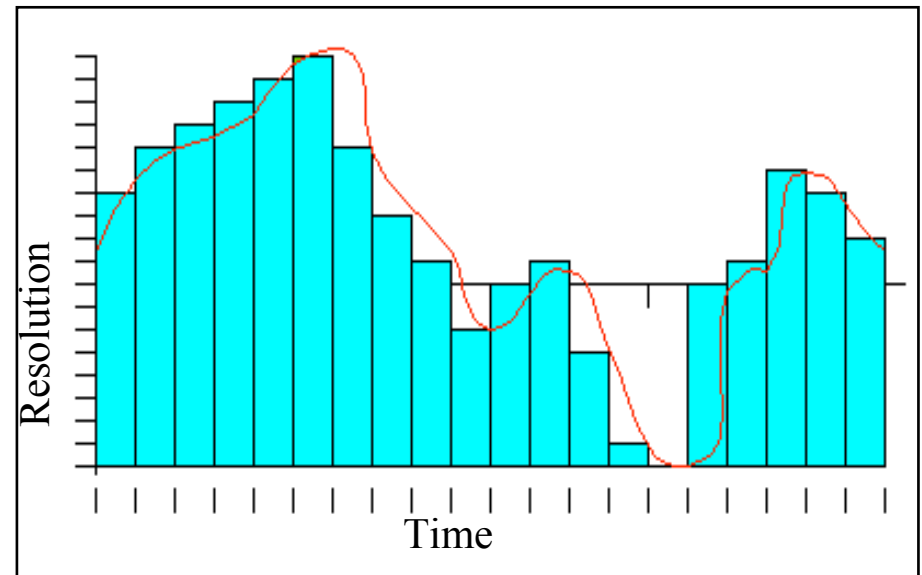
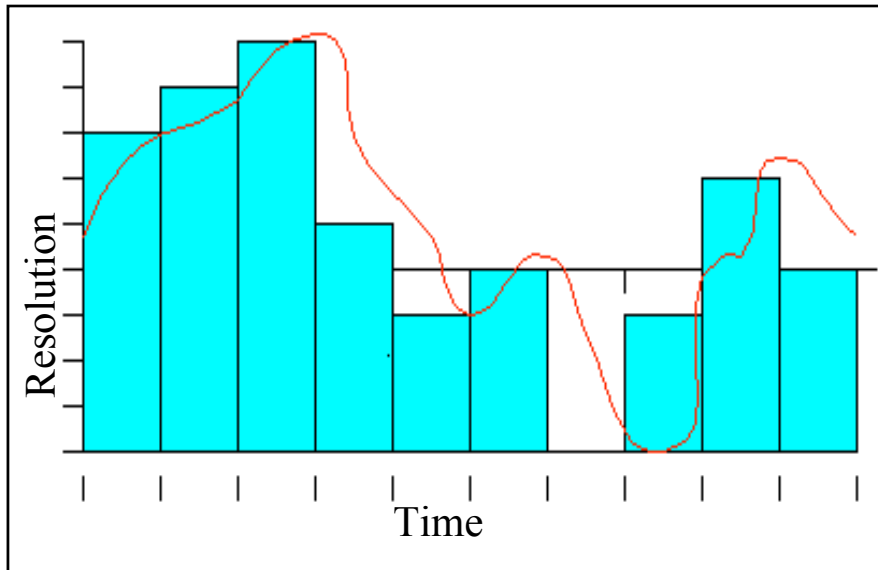
There are two ways to best improve the accuracy of A/D conversion:

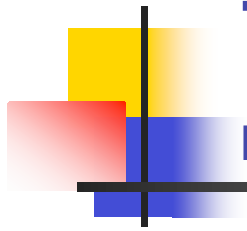
- increasing the resolution which improves the accuracy in measuring the amplitude of the analog signal.
- increasing the sampling rate which increases the maximum frequency that can be measured.

# Accuracy of A/D Conversion

■ Low Accuracy

■ Improved





# Types of A/D Converters

By: Todd Sifleet

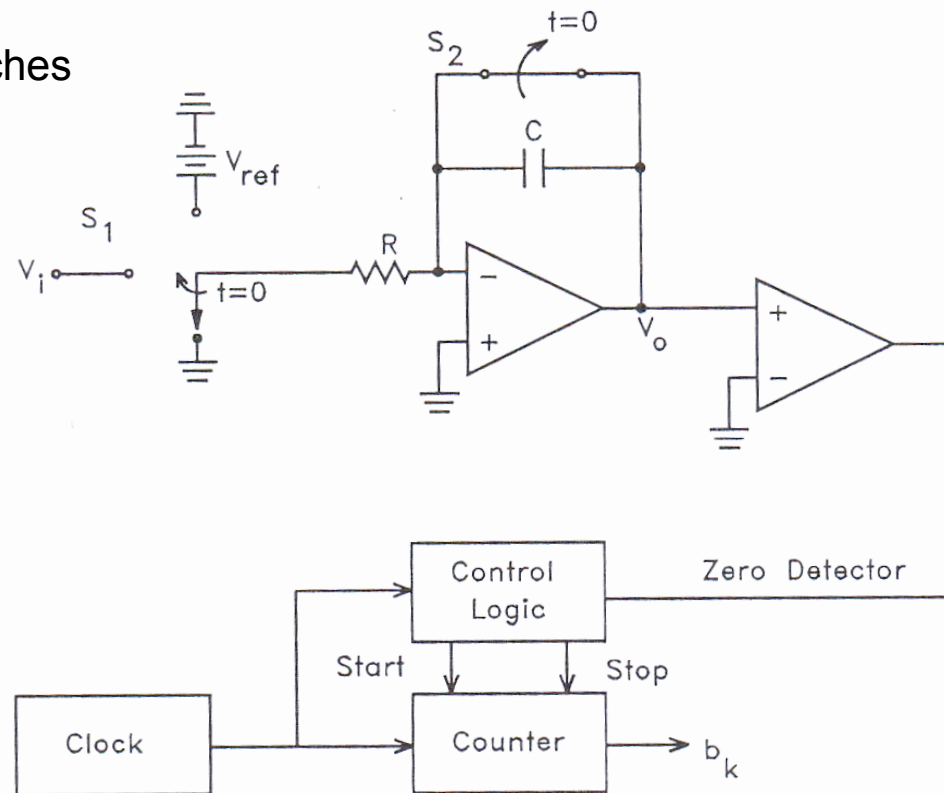
---

- Dual Slope A/D Converter
- Successive Approximation A/D Converter
- Flash A/D Converter
- Delta-Sigma A/D Converter
- Other
  - Voltage-to-frequency, staircase ramp or single slope, charge balancing or redistribution, switched capacitor, tracking, and synchro or resolver

# Dual Slope A/D Converter

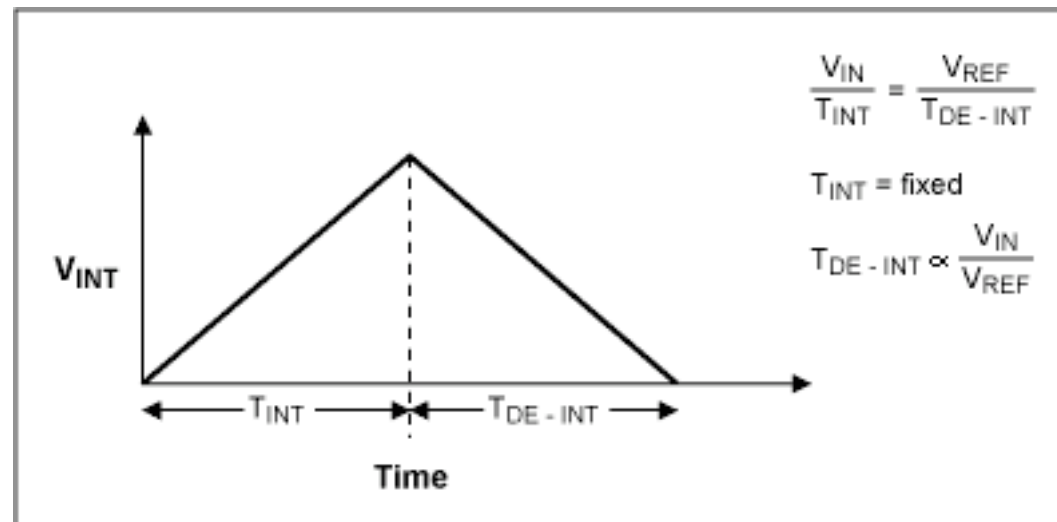
- Fundamental components

- Integrator
- Electronically Controlled Switches
- Counter
- Clock
- Control Logic
- Comparator



# How does it work

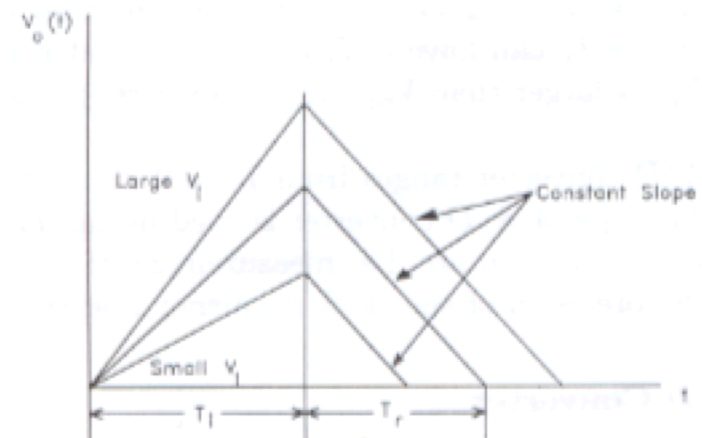
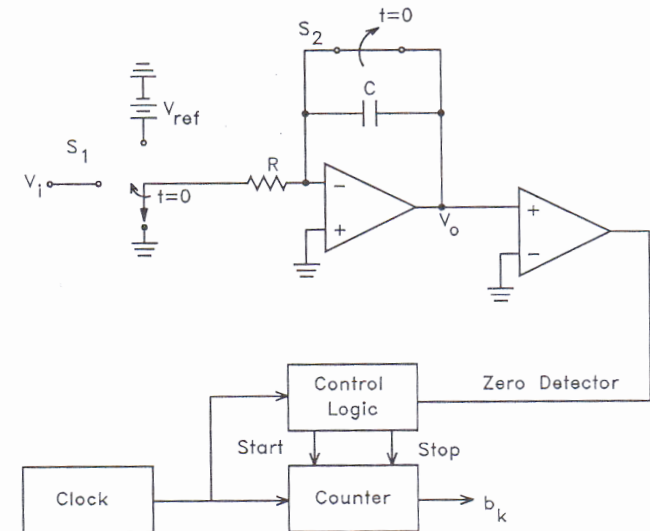
A dual-slope ADC (DS-ADC) integrates an unknown input voltage ( $V_{IN}$ ) for a fixed amount of time ( $T_{INT}$ ), then "de-integrates" ( $T_{DEINT}$ ) using a known reference voltage ( $V_{REF}$ ) for a variable amount of time.



The key advantage of this architecture over the single-slope is that the final conversion result is insensitive to errors in the component values. That is, any error introduced by a component value during the integrate cycle will be cancelled out during the de-integrate phase.

# How Does it Work Cont.

- At  $t < 0$ ,  $S_1$  is set to ground,  $S_2$  is closed, and counter=0.
- At  $t=0$  a conversion begins and  $S_2$  is open, and  $S_1$  is set so the input to the integrator is  $V_{in}$ .
- $S_1$  is held for  $T_{INT}$  which is a constant predetermined time interval.
- When  $S_1$  is set the counter begins to count clock pulses, the counter resets to zero after  $T_{INT}$
- $V_{out}$  of integrator at  $t=T_{INT}$  is  $V_{IN} T_{INT}/RC$  is linearly proportional to  $V_{IN}$
- At  $t=T_{INT}$   $S_1$  is set so  $-V_{ref}$  is the input to the integrator which has the voltage  $V_{IN} T_{INT}/RC$  stored in it.
- The integrator voltage then drops linearly with a slope  $-V_{ref}/RC$ .
- A comparator is used to determine when the output voltage of the integrator crosses zero
- When it is zero the digitized output value is the state of the counter.





# Dual Slope A/D Converter

## Pros and Cons

---

### PROS

- Conversion result is insensitive to errors in the component values.
- Fewer adverse affects from “noise”
- High Accuracy

### CONS

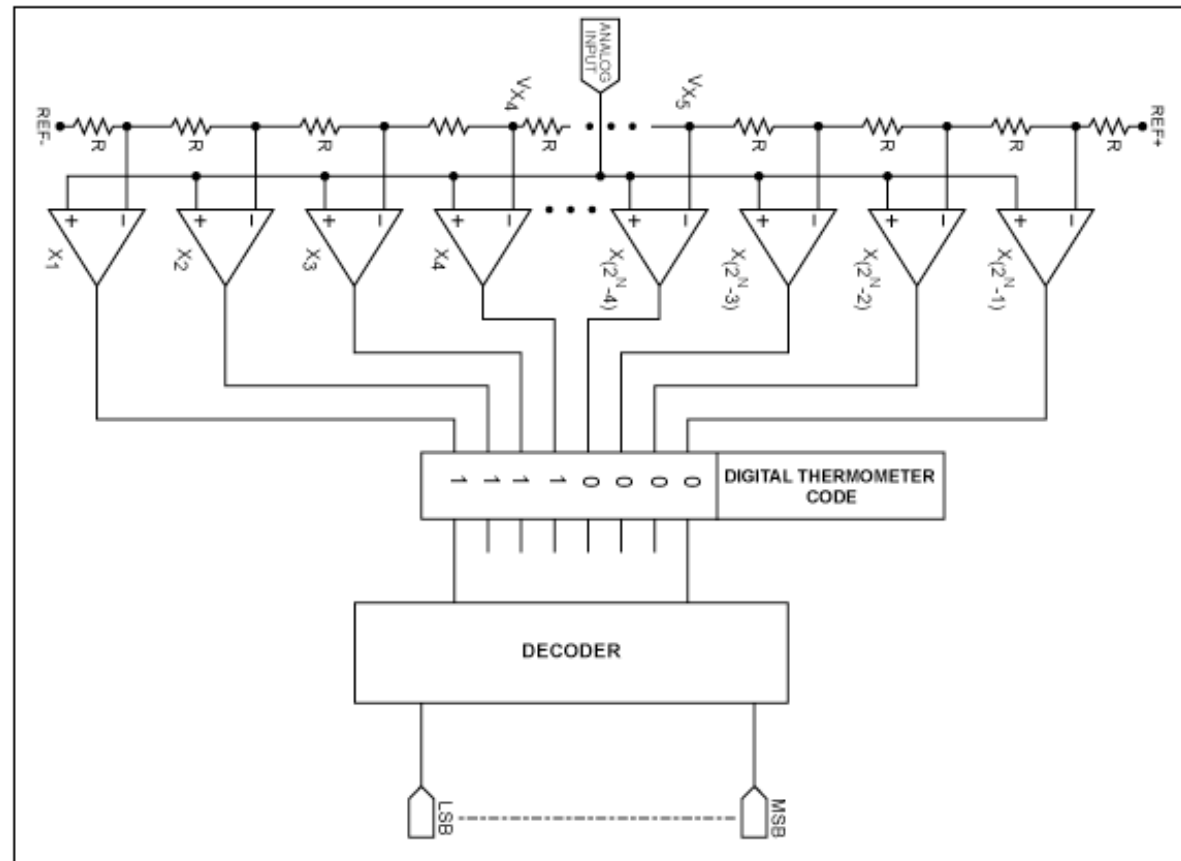
- Slow
- Accuracy is dependent on the use of precision external components
- Cost

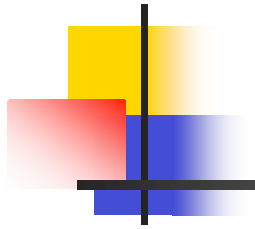


# Flash A/D Converter

- Fundamental Components (For N bit Flash A/D)

- $2^N - 1$  Comparators
- $2^N$  Resistors
- Control Logic





# How does it work

---

- Uses the  $2^N$  resistors to form a ladder voltage divider, which divides the reference voltage into  $2^N$  equal intervals.
- Uses the  $2^N-1$  comparators to determine in which of these  $2^N$  voltage intervals the input voltage  $V_{in}$  lies.
- The Combinational logic then translates the information provided by the output of the comparators
- This ADC does not require a clock so the conversion time is essentially set by the settling time of the comparators and the propagation time of the combinational logic.



# Flash A/D Converter

## Pros and Cons

---

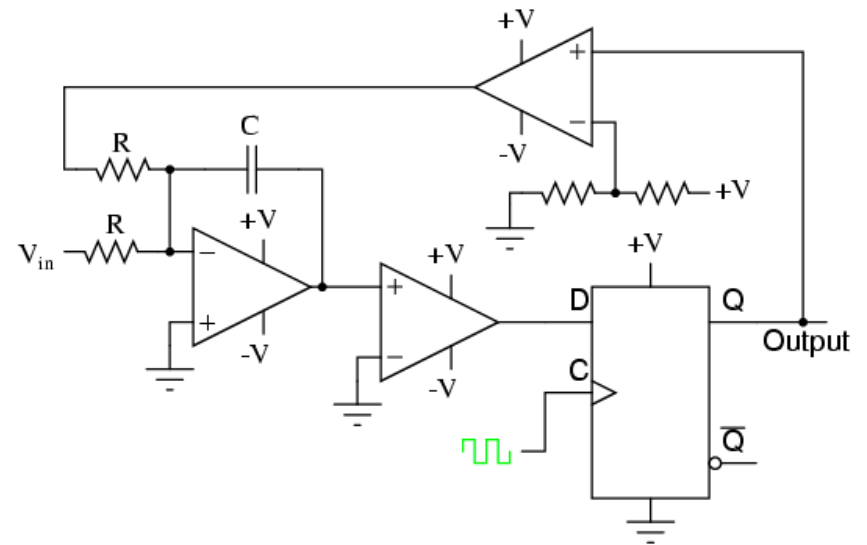
### PROS

- Very Fast (Fastest)
- Very simple operational theory
- Speed is only limited by gate and comparator propagation delay

### CONS

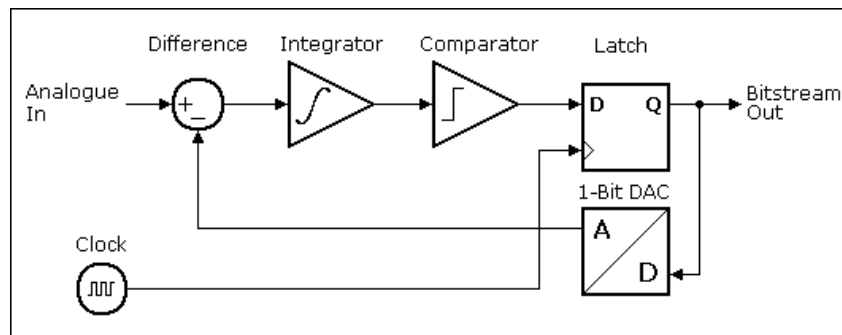
- Expensive
- Prone to produce glitches in the output
- Each additional bit of resolution requires twice the comparators.

- Resistors
- Capacitor
- Comparators
- Control Logic
- DAC



# How does it work

- Input is over sampled, and goes to integrator.
- The integration is then compared to ground.
- Iterates and produces a serial bit stream
- Output is a serial bit stream with # of 1's proportional to  $V_{in}$



- With this arrangement the sigma-delta modulator automatically adjusts its output to ensure that the average error at the quantizer output is zero.
- The integrator value is the sum of all past values of the error, so whenever there is a non-zero error value the integrator value just keeps building until the error is once again forced to zero.



# Sigma-Delta A/D Converter

## Pros and Cons

---

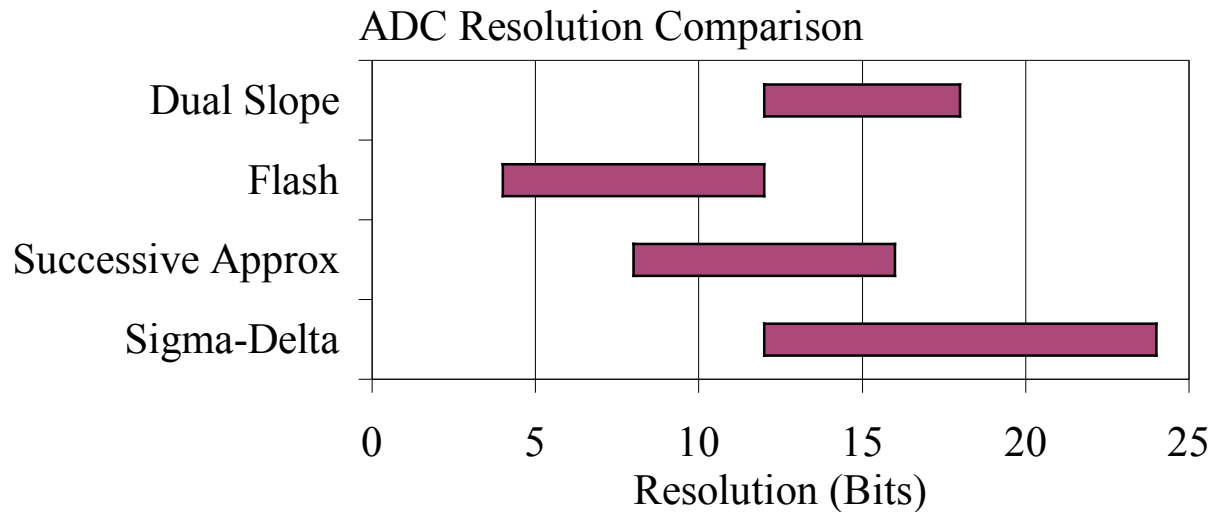
### PROS

- High Resolution
- No need for precision components

### CONS

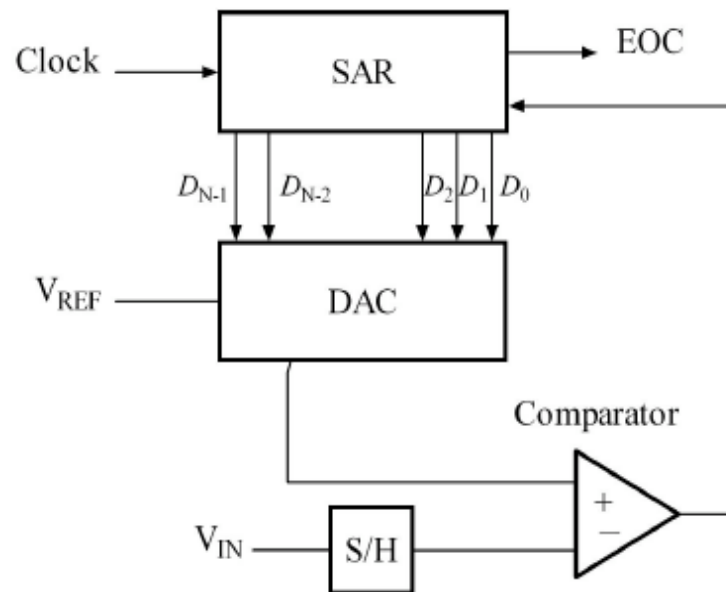
- Slow due to over sampling
- Only good for low bandwidth

# ADC Types Comparison



Type	Speed (relative)	Cost (relative)
Dual Slope	Slow	Med
Flash	Very Fast	High
Successive Appox	Medium – Fast	Low
Sigma-Delta	Slow	Low

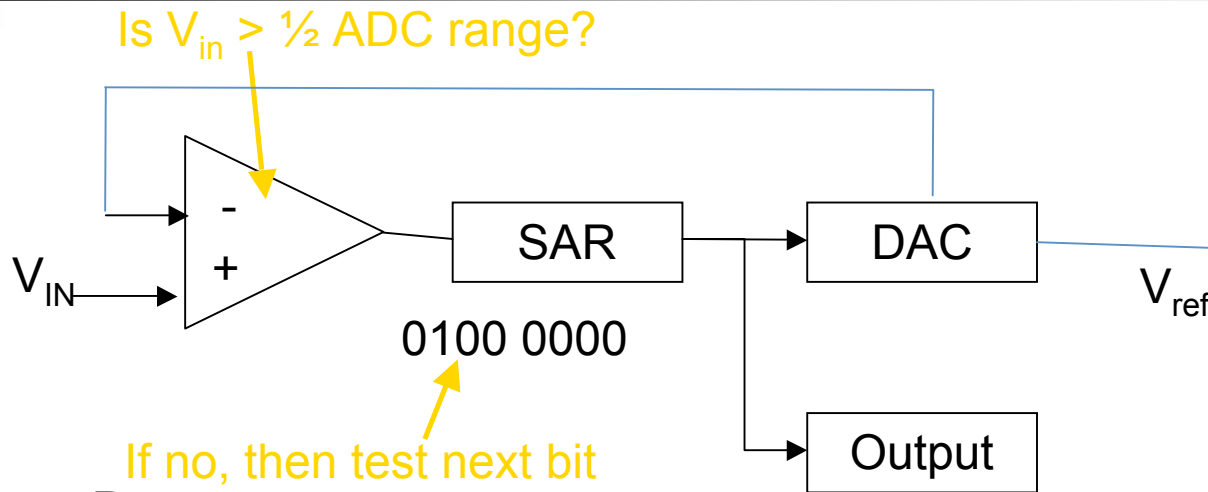
# Successive Approximation ADC Circuit



- Uses a n-bit DAC to compare DAC and original analog results.
- Uses Successive Approximation Register (SAR) supplies an approximate digital code to DAC of  $V_{IN}$ .
- Comparison changes digital output to bring it closer to the input value.
- Uses Closed-Loop Feedback Conversion



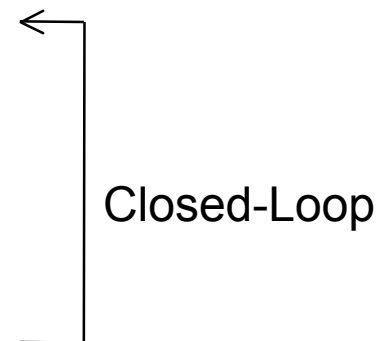
# Successive Approximation ADC



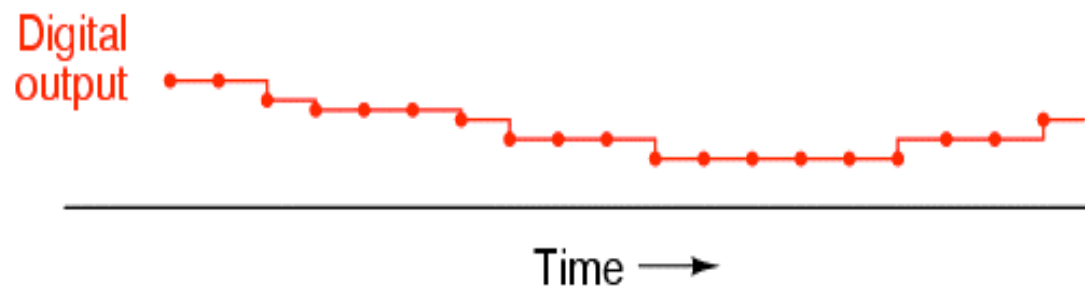
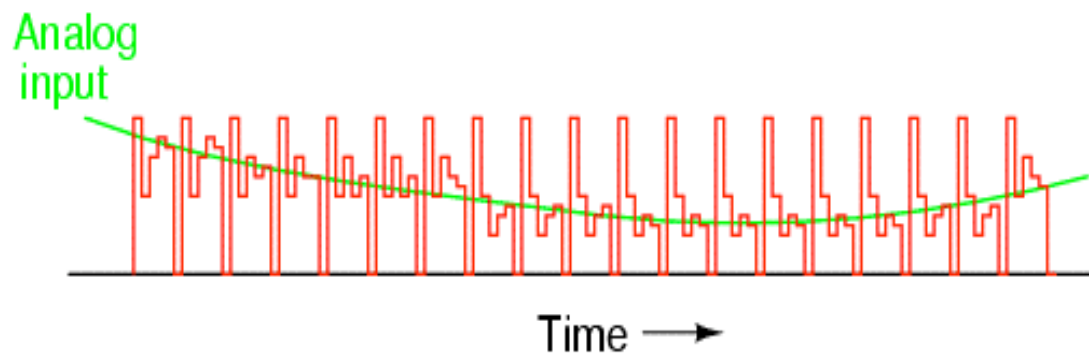
If no, then test next bit

## Process

1. MSB initialized as 1
2. Convert digital value to analog using DAC
3. Compares guess to analog input
4. Is  $V_{in} > V_{DAC}$ 
  - Set bit 1
  - If no, bit is 0 and test next bit



# Output





# Successive Approximation

## Disadvantages

## Advantages

- Capable of high speed and reliable
  - Medium accuracy compared to other ADC types
  - Good tradeoff between speed and cost
  - Capable of outputting the binary number in serial (one bit at a time) format.
- Higher resolution successive approximation ADC's will be slower
  - Speed limited to ~5Msps

# Successive Approximation Example

## Example

- 10 bit ADC
- $V_{in} = 0.6$  volts (from analog device)
- $V_{ref} = 1$  volts
- Find the digital value of  $V_{in}$

Bit	Voltage
9	.5
8	.25
7	.125
6	.0625
5	.03125
4	.015625
3	.0078125
2	.00390625
1	.001952125
0	.0009765625

$N = 2^n$  (N of possible states)

$N = 1024$

$V_{max} - V_{min} / N = 1 \text{ Volt} / 1024 =$   
 $0.0009765625V$  of  $V_{ref}$  (resolution)

- |   |  |  |  |  |  |  |  |  |  |
|---|--|--|--|--|--|--|--|--|--|
| 1 |  |  |  |  |  |  |  |  |  |
|---|--|--|--|--|--|--|--|--|--|

Bit	Voltage
9	.5
8	.25
7	.125
6	.0625
5	.03125
4	.015625
3	.0078125
2	.00390625
1	.001952125
0	.0009765625



# Successive Approximation

- Next Calculate MSB-1 (bit 8)
  - Compare  $V_{in}=0.6\text{ V}$  to  $V=V_{ref}/2 + V_{ref}/4= 0.5+0.25 =0.75\text{V}$
  - Since  $0.6<0.75$ , MSB is turned off
- Calculate MSB-2 (bit 7)
  - Go back to the last voltage that caused it to be turned on (Bit 9) and add it to  $V_{ref}/8$ , and compare with  $V_{in}$
  - Compare  $V_{in}$  with  $(0.5+V_{ref}/8)=0.625$
  - Since  $0.6<0.625$ , MSB is turned off

1	0	0							
---	---	---	--	--	--	--	--	--	--



# Successive Approximation

- Calculate the state of MSB-3 (bit 6)
  - Go to the last bit that caused it to be turned on (In this case MSB-1) and add it to  $V_{\text{ref}}/16$ , and compare it to  $V_{\text{in}}$
  - Compare  $V_{\text{in}}$  to  $V = 0.5 + V_{\text{ref}}/16 = 0.5625$
  - Since  $0.6 > 0.5625$ , MSB-3=1 (turned on)

MSB	MSB-1	MSB-2	MSB-3	...					
1	0	0	1						

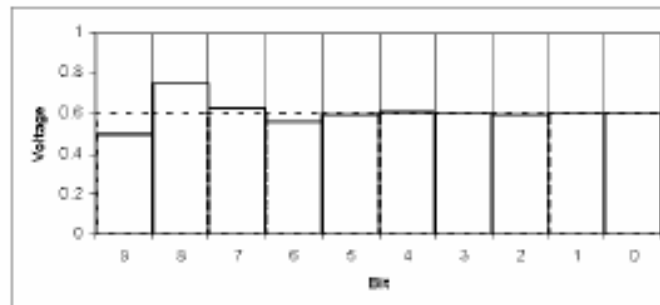
# Successive Approximation ADC

- This process continues for all the remaining bits.

•Digital Results:

MSB	MSB-1	MSB-2	MSB-3	...					LSB
1	0	0	1	1	0	0	1	1	0

•Results:  $\frac{1}{2} + \frac{1}{16} + \frac{1}{32} + \frac{1}{256} + \frac{1}{512} = .599609375 \text{ V}$



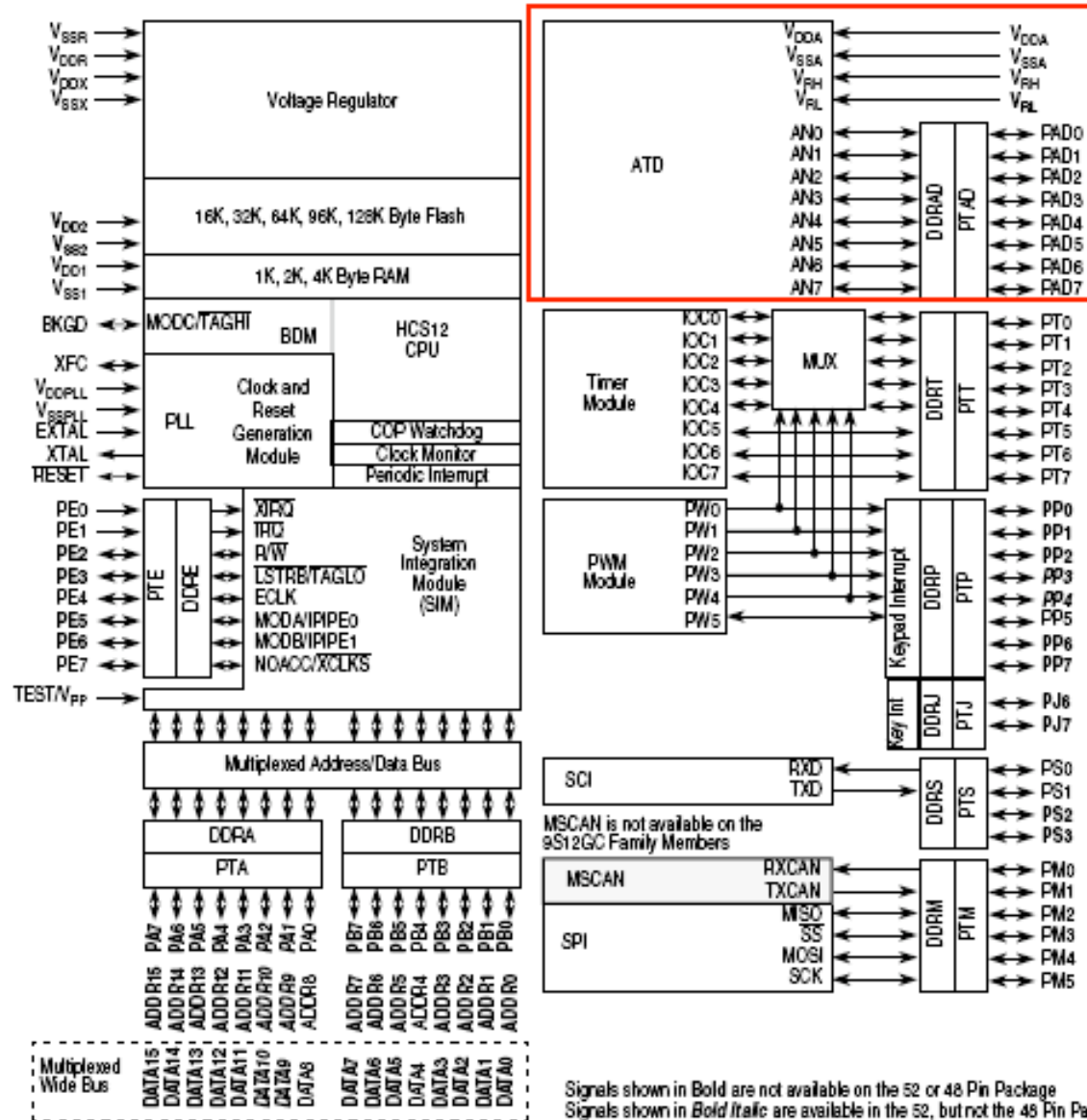
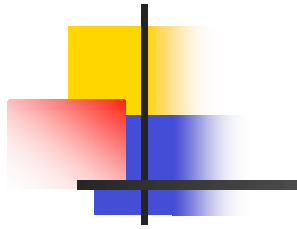




# ATD10B8CV2 on MC9SI2C

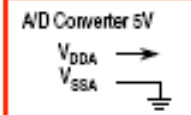
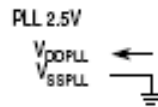
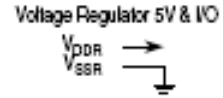
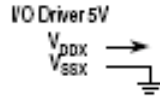
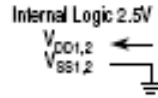
By: Yuchun Huang

---



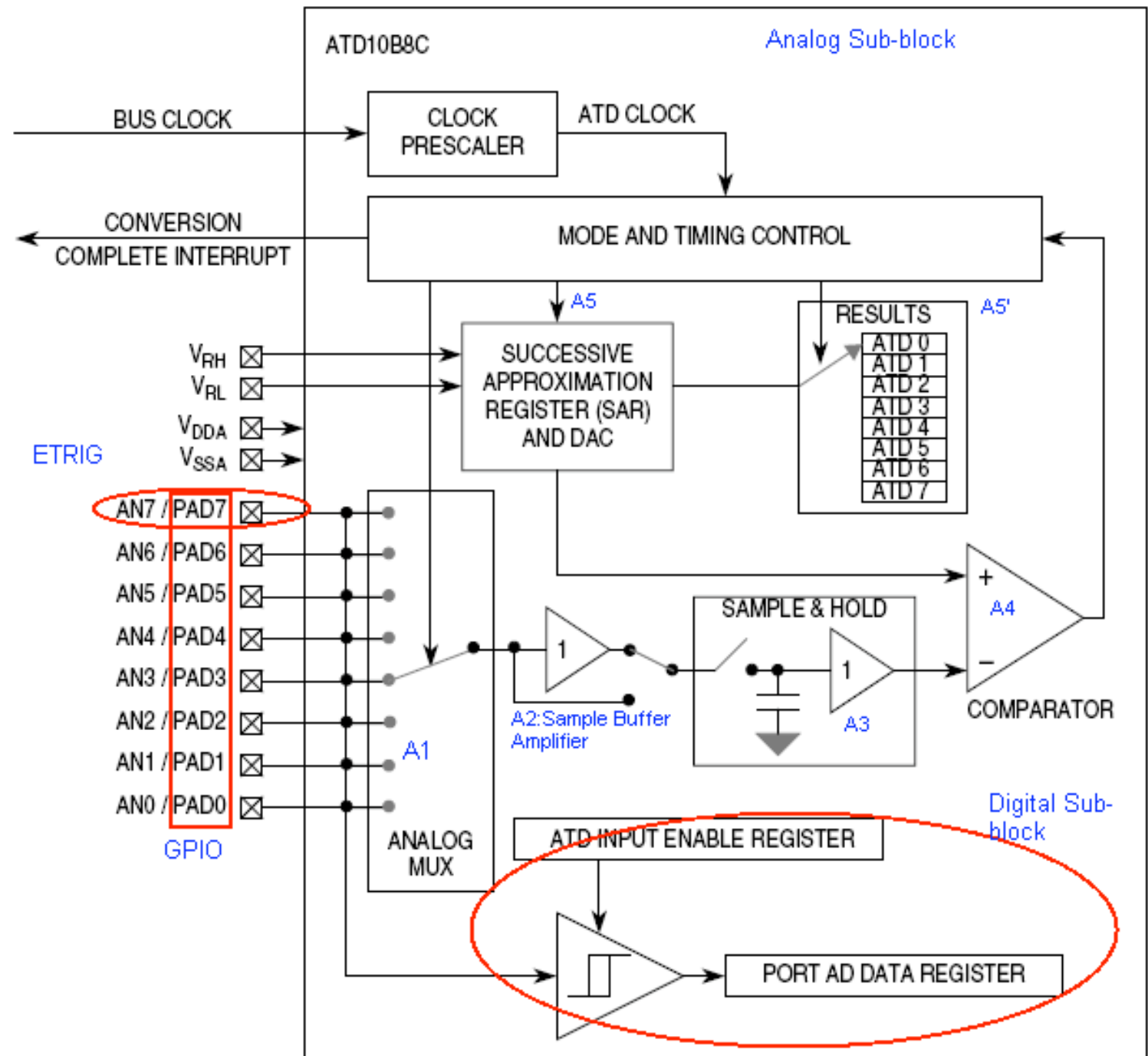
AD  
C

ADC Power  
Supply



$V_{RL}$  is bonded internally to  $V_{SSA}$  for 52- and 48-Pin packages

Signals shown in Bold are not available on the 52 or 48 Pin Package  
Signals shown in *Bold Italic* are available in the 52, but not the 48 Pin Package





# ATD10B8CV2 Features

---

- Resolution:
  - 8/10-bit
- Conversion Time:
  - 7  $\mu$ sec for 10-bit
- Successive Approximation ADC
- 8-channel analog/digital input multiplexer
  - Multiplexer: A device that can send several signals over a single line.
- External Trigger Control (Sync. ADC with external events)
- New Features of ATD10B8CV2
  - Conversion Complete Interrupt
  - Left/right justified, signed/unsigned result
  - Programmable Sample time & resolution selection



# ATD10B8CV2

## Modes & Operations

---

- **Modes**

- Stop Mode
  - All clocks halt; conversion aborts; minimum recovery delay
- Wait Mode
  - Reduced MCU power; can resume
- Freeze Mode
  - Breakpoint for debugging an application

- **Operations**

- Setting up and Starting the A/D Conversion
- Aborting the A/D Conversion
- Resets
- Interrupts

# ATD10B8CV2

## Registers & Memory Map

---

- Number of Control Register is 6: 0x0000~0x0005
  - ATDCTL2    0x0080+0x0002
    - Power down; external trigger; interrupts
  - ATDCTL3    0x0080+0x0003
    - Conversion Sequence Length; FIFO for result register; behavior in Freeze mode
  - ATDCTL4    0x0080+0x0004
    - Resolution; conversion clock freq.; sample time
  - ATDCTL5    0x0080+0x0005
    - Type of conversion sequence; analog input channel selection



# ATD10B8CV2

## Registers&Memory Map

---

- Number of Status Register is 2: 0x0006 / 0x000B
  - ATDSTAT0    0x0080+0x0006
    - Power down; external trigger; interrupts
  - ATDSTAT1    0x0080+0x000B
    - Conversion Sequence Length; FIFO for result register; behavior in Freeze mode
- Number of Special/Test Register is 1: 0x0009
  - ATDTEST1    0x0080+0x0009
    - Power down; external trigger; interrupts
- Number of Conversion Result Registers is 16: 0x0010~0x001F
  - ATDDR0H~ATDDR7L:    0x0080+0x0010~0x001F
    - Power down; external trigger; interrupts



# ATD10B8CV2

## Registers & Memory Map

---

- Digital Input Enable Register # is 1: 0x000D
  - ATDDIEN 0x0080+0x000D
    - Power down; external trigger; interrupts
  - ATDSTAT1 0x0080+0x000B
    - Conversion Sequence Length; FIFO for result register; behavior in Freeze mode
- Digital Port Data Register # is 1: 0x000F
  - PORTAD 0x0080+0x000F
    - Power down; external trigger; interrupts
- Total # registers in ATD is 32: 0x0000~0x001F
  - Including unimplemented or reserved ones:



# ATD10B8CV2 - Control Registers

## • ATDCTL2

\$0082

A/D power down; external trigger; interrupt

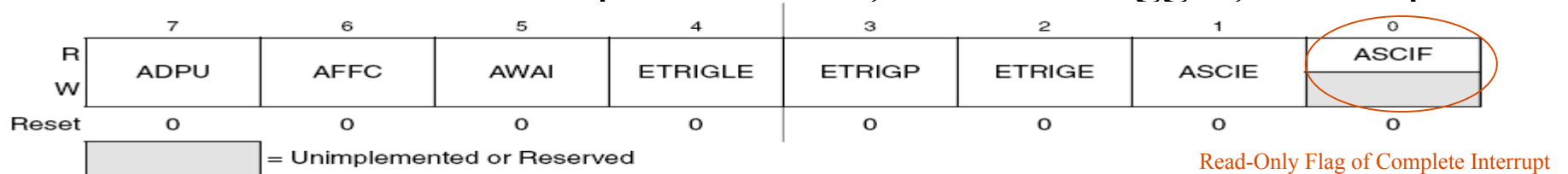


Figure 8-5. ATD Control Register 2 (ATDCTL2)

Read: Anytime  
Write: Anytime

Abort the  
current A/D  
sequence

Not start a  
new sequence

PIN	Description
7	0-power down ATD; recovery time period 1-normal ATD functionality
6	0-normal clearing (read the CCF before reading the result register ) 1-Fast Flag Clearing (automatically clear CCF after any access to result register)
5	0-continue to run in Wait Mode; 1-Halt conversion and power down ATD during Wait Mode
4	0-External Trigger Edge 1-Trigger Level
3	0-Low /Falling trigger polarity 1-high/Rising trigger polarity
2	0-Disable external trigger mode 1-external trigger mode Enabled (Channel AN7)
1	0-ATD Sequence Complete Interrupt Request Disabled 1-ATD Sequence Complete Interrupt Request Enabled

# ATD10B8CV2 - Control Registers

## • ATDCTL3

\$0083

Sequence Length; FIFO Result; Freeze Behavior

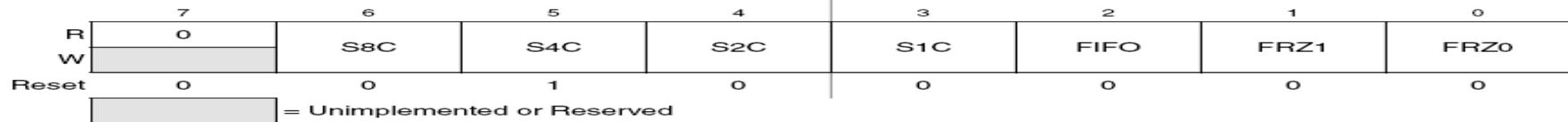


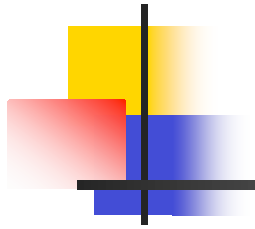
Figure 8-6. ATD Control Register 3 (ATDCTL3)

Read: Anytime  
Write: Anytime

Abort the  
current A/D  
sequence

Not start a  
new sequence

PIN	Description
6~3	# of conversions per sequence (Table 8-4, P233) [1~8]
2	0- A/D Conversion #1-→result register #1,.....( <u>fixed</u> ) 1- current conversion-→ conversion counter result register, <u>sequentially</u> , <u>wrap</u> around when reaching maximum register # *: starting a new conversion by writing to ATDCTL5 clear the conversion counter to Zero -→first conversion---always in first result register ATDDR0
1	Breakpoint Behavior (Freeze Mode) when debugging (Table 8-5, P233) [00:continue conversion;10:finish current conversion then freeze;11:freeze immediattly ]
0	



# ATD10B8CV2 - Control Registers

## • ATDCTL4

\$0084

Resolution; Sample Time

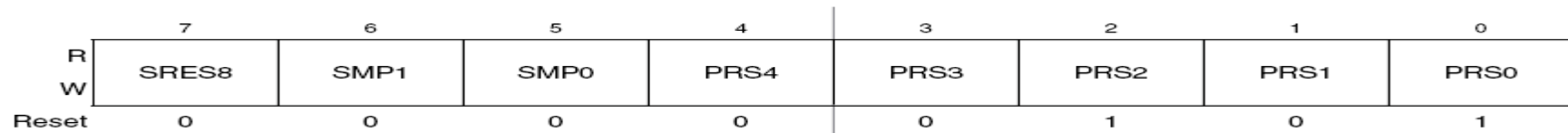


Figure 8-7. ATD Control Register 4 (ATDCTL4)

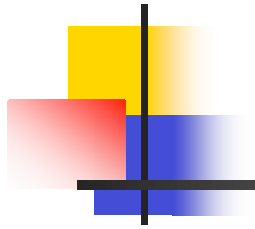
Read: Anytime

Write: Anytime

Abort the  
current A/D  
sequence

Not start a  
new sequence

PIN	Description
7	Resolution [0:10-bit; 1:8-bit ]
6~5	Second phase of Sample time (Table 8-8, P237) [00: 2 clock period; 01: 4 ; 10: 8; 11: 16]
4~0	A/D Clock Prescaler [PRS: 0 ~31] $ATDclock = \frac{[BusClock]}{[PRS + 1]} \times 0.5$



# ATD10B8CV2 - Control Registers

- **ATDCTL5**

\$0085

Type of conversion; sampled channels

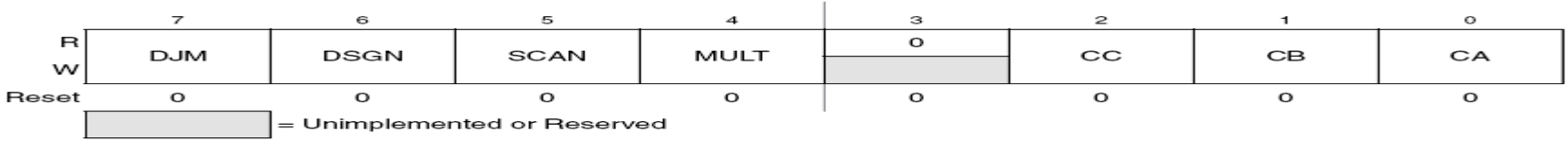


Figure 8-8. ATD Control Register 5 (ATDCTL5)

Read: Anytime  
Write: Anytime

Abort the  
current A/D  
sequence

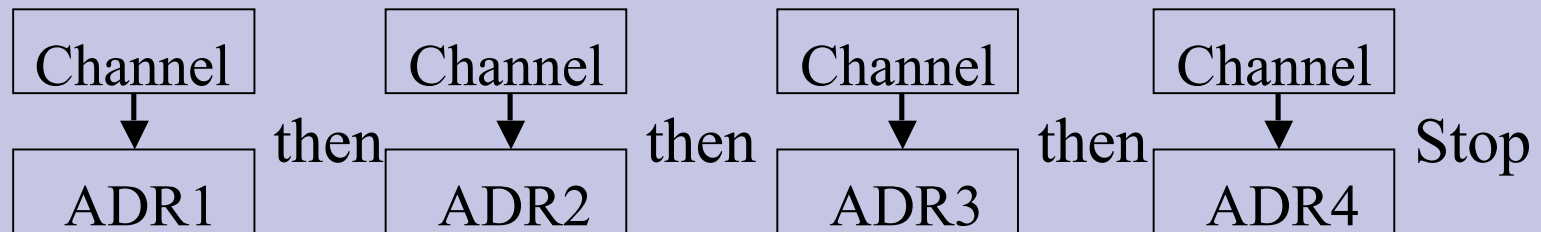
Start a new  
sequence

PIN	Description
7	0-Result Register Justification Mode : Left X---15 both signed and unsigned 1-Result Register Justification Mode : Right 0---X unsigned only
6	0- Unsigned data in result register 1- Signed data in result register [2's complement]
5	0- Single Conversion Mode; Only Once 1- Continuous Conversion Mode. Scanning
4	0- Sample Only one channel [Selected by CC~CA, # by S8C~S1C in ATDCTL3] 1- Sample across several channels [first by CC~CA, increase the subseq. Channel code]
2~0	Analog Channel Input Selection Code (Table 8-12, P239) [000: AN0; 001:AN1; ..... ; 111:AN7 MULT=0: selected channel; MULT=1: first channel in sequence, then sequentially, auto-wrap]

# Single Channel, 4-Conversion Sequence

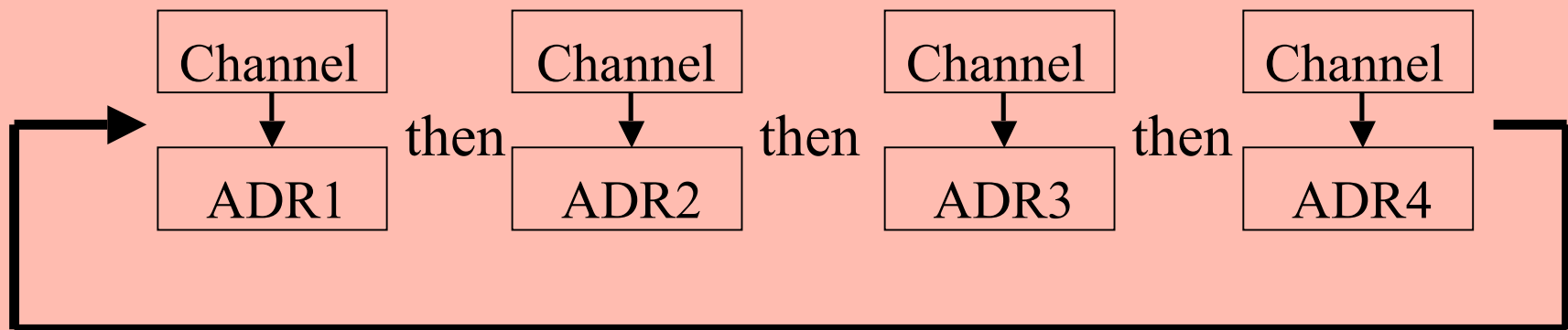
## CONSECUTIVE

Set MULT = 0 (for single channel) and SCAN = 0 (for consecutive)



## CONTINUOUS

Set MULT = 0 (for single channel) and SCAN = 1 (for continuous)

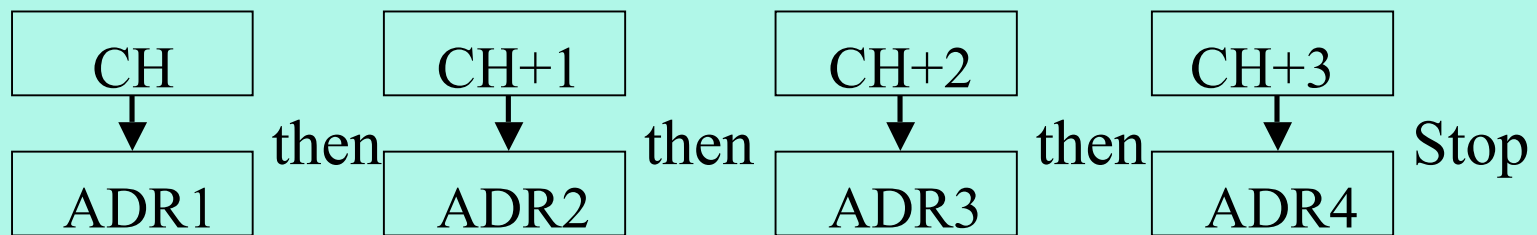


Overwrite ADR's

# Multi Channel, 4-Conversion Sequence

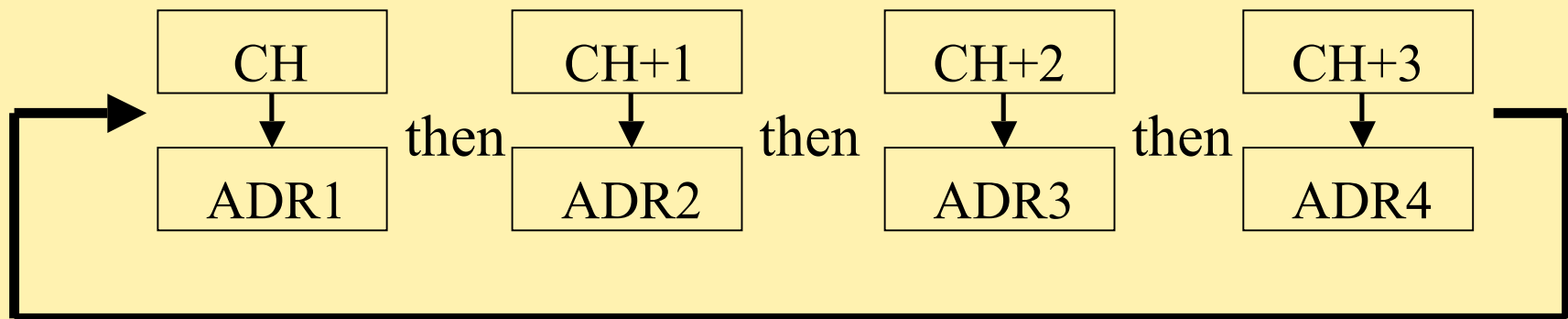
## CONSECUTIVE

Set MULT = 1 (for multi-channel) and SCAN = 0 (for consecutive)



## CONTINUOUS

Set MULT = 1 (for multi-channel) and SCAN = 1 (for continuous)



Overwrite ADR's

# ATD10B8CV2 - Status Registers

## • ATDSTAT0

\$0086

Sequence Complete Flag; Overrun; Conversion Counter

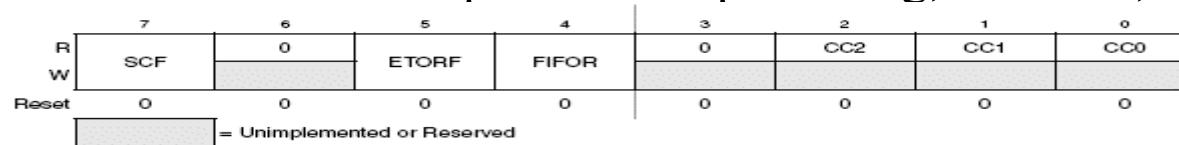


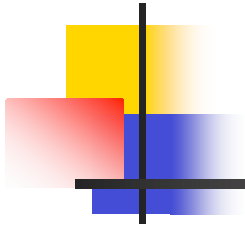
Figure 8-9. ATD Status Register 0 (ATDSTAT0)

Read: Anytime

Write: Anytime (no effect on (CC2, CC1, CC0))

Read-only

PIN	Description
7	0- Conversion Sequence not completed; 1- Conversion Sequence completed. [SCAN=1 set after each completed seq.]
5	0- No external trigger overrun error has occurred; 1- external trigger overrun error has occurred [ <i>ETrigLE=0, additional active edge while a conversion sequence is in progress</i> ]
4	0- No overrun in results; 1- An overrun in results. [ <i>result registers written before its CCF is cleared. Old data lost before reading</i> ]
3~0	Conversion counter [*: The conversion counter points to the result register that will receive the result of the current conversion; *: FIFO=0, the conversion counter is initialized to zero at the begin and end of the conversion sequence; *: FIFO=1, the register counter is not initialized. The conversion counters wraps around when its maximum value is reached.]



# ATD10B8CV2-Status Registers

- **ATDSTAT1**  
**\$008B**

## Conversion Complete Flag

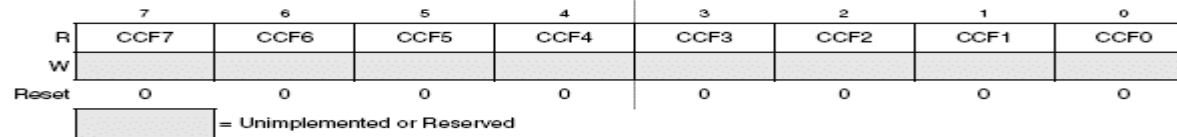


Figure 8-12. ATD Status Register 1 (ATDSTAT1)

Read: Anytime

Write: Anytime, no effect

Read-only

PIN	Description
7~0	0- conversion # x is not completed; 1- conversion # x is completed, results available. <i>[set at the end of each conversion in a sequence]</i>



# ATD10B8CV2 -Result Registers

- **ATDDRH<sub>x</sub>/ATDDRL<sub>x</sub>**  
\$0090~\$009F      8 result registers, read-only

## 8.3.2.13.1 Left Justified Result Data

Module Base + 0x0010 = ATDDR0H, 0x0012 = ATDDR1H, 0x0014 = ATDDR2H, 0x0016 = ATDDR3H  
0x0018 = ATDDR4H, 0x001A = ATDDR5H, 0x001C = ATDDR6H, 0x001E = ATDDR7H

	7	6	5	4	3	2	1	0	
R	BIT 9 MSB	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	10-bit data
W	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	8-bit data
Reset	0	0	0	0	0	0	0	0	

Figure 8-15. Left Justified, ATD Conversion Result Register, High Byte (ATDDRH)

Module Base + 0x0011 = ATDDR0L, 0x0013 = ATDDR1L, 0x0015 = ATDDR2L, 0x0017 = ATDDR3L  
0x0019 = ATDDR4L, 0x001B = ATDDR5L, 0x001D = ATDDR6L, 0x001F = ATDDR7L

	7	6	5	4	3	2	1	0	
R	BIT 1	BIT 0	0	0	0	0	0	0	10-bit data
W	U	U	0	0	0	0	0	0	8-bit data
Reset	0	0	0	0	0	0	0	0	

Figure 8-16. Left Justified, ATD Conversion Result Register, Low Byte (ATDDRL)

## 8.3.2.13.2 Right Justified Result Data

Module Base + 0x0010 = ATDDR0H, 0x0012 = ATDDR1H, 0x0014 = ATDDR2H, 0x0016 = ATDDR3H  
0x0018 = ATDDR4H, 0x001A = ATDDR5H, 0x001C = ATDDR6H, 0x001E = ATDDR7H

	7	6	5	4	3	2	1	0	
R	0	0	0	0	0	0	BIT 9 MSB	BIT 8	10-bit data
W	0	0	0	0	0	0	0	0	8-bit data
Reset	0	0	0	0	0	0	0	0	

Figure 8-17. Right Justified, ATD Conversion Result Register, High Byte (ATDDRH)

Module Base + 0x0011 = ATDDR0L, 0x0013 = ATDDR1L, 0x0015 = ATDDR2L, 0x0017 = ATDDR3L  
0x0019 = ATDDR4L, 0x001B = ATDDR5L, 0x001D = ATDDR6L, 0x001F = ATDDR7L

	7	6	5	4	3	2	1	0	
R	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	10-bit data
W	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	8-bit data
Reset	0	0	0	0	0	0	0	0	

Figure 8-18. Right Justified, ATD Conversion Result Register, Low Byte (ATDDRL)

# ATD10B8CV2 -Test/Special Registers

## • ATDCTL2 \$0089

### Special Channel conversion

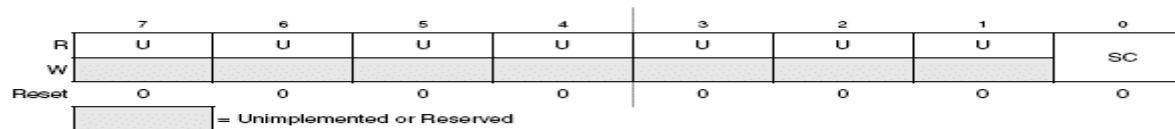


Figure 8-11. ATD Test Register 1 (ATDTEST1)

Read: Anytime, returns unpredictable values for Bit 7 and Bit 6

Write: Anytime

PIN	Description
0	<p>0- special channel conversion disabled; 1- special channel conversion selected.</p> <p><b>[Note:</b> Always write remaining bits of ATDTEST1 (Bit7 to Bit1) zero when writing SC bit. Not doing so might result in unpredictable ATD behavior; *: special meaning for CC~CA in ATDCTL5]</p>

Table 8-15. Special Channel Select Coding

SC	CC	CB	CA	Analog Input Channel
1	0	X	X	Reserved
1	1	0	0	V <sub>RH</sub>
1	1	0	1	V <sub>RL</sub>
1	1	1	0	(V <sub>RH</sub> +V <sub>RL</sub> ) / 2
1	1	1	1	Reserved

# ATD10B8CV2 - GPIO Registers

- **ATDDIEN**

\$008D

Digital Input enabled flag from analog pin AN0~AN7 to PTADx registers

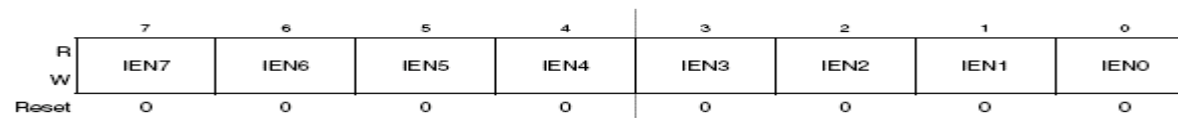


Figure 8-13. ATD Input Enable Register (ATDDIEN)

Read: Anytime

Write: Anytime

PIN	Description
7~0	0- disable digital input buffer from analog pin ANx to PTADx registers; 1- enable digital input buffer from analog pin ANx to PTADx registers. <b>[Note:</b> If this bit is set while simultaneously using it as an analog port, there is <i>potentially increased power consumption</i> <b>]</b>

# ATD10B8CV2 - GPIO Registers

- **PORTAD**  
\$008F

## GPIO port data registers

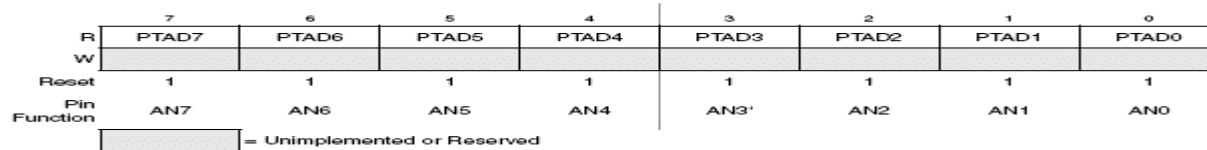


Figure 8-14. Port Data Register (PORTAD)

Read: Anytime

Write: Anytime, no effect

PIN	Description
7~0	If IENx=1, Read the logic level on ANx pin [Note: signal potentials not meeting VIL or VIH specifications will have an indeterminate value]



# ATD10B8CV2 - Setting up and Starting

---

- Step 1:
  - Power up the ATD and define settings in ATDCTL2
    - ADPU-→1 (power up) ; ASCIE=1 (enable interrupt)
- Step 2:
  - Wait for the ATD recovery time before proceeding
    - CPU loop
- Step 3:
  - Configure number of conversion in a sequence
    - S8C~S1C , **FIFO** in ATDCTL3
- Step 4:
  - Configure resolution, sampling time and ATD clock Speed
    - SRES8; SMP1~0; PRS4~0 in ATDCTL4
- Step 5:
  - Configure Starting Channel, Single/Multiple Channel, SCAN and result data SIGNED?
    - CC~CA, MULT, SCAN, DJM, DSGN in ATDCTL5



## ATD10B8CV2 - Aborting the conversion

---

- **Step 1:**
  - **Disable the ATD Interrupt in ATDCTL2**
    - **ASCIE→0 (disable interrupt)**
- **Step 2:**
  - **Clear SCF flag in ATDSTAT0**
    - **SCF-→1**
- **Step 3:**
  - **Power down ATD**
    - **ADPU=0 in ATDCTL2**

# Using the ATD10B8C to Read Chan. AN0

```
ADCTL2    EQU    $0082
ADCTL3    EQU    $0083
ADCTL4    EQU    $0083
ADCTL5    EQU    $0084
ADSTAT    EQU    $0086
ADR1      EQU    $1031
ADRESULT  RMB    1
```

```

                                ;ADPU=1,ASCIE=1
                                ;  "
                                ;S4C=1
                                ;  "
                                ;8-bit, /10
                                ;
                                ;delay for 105 µs
DELAY      LDY    #30
            DEY
            BNE   DELAY

            LDAA   #$10          ;Left unsigned, SCAN=0,MULT=1,CHAN=000
            STAA   ADCTL5        ; start conversion
            LDX    #ADSTAT       ;check for complete flag
            BRCLR  0,X,$80,*     ;SCF is bit 7

            LDAA   ADR1          ;read chan. 0
            STAA   ADRESULT      ;store in result
            SWI
```

Turn on charge pump and  
select clock source

Delay for charge pump to  
stabilize

Set ADCTL5 to start  
conversion

Wait until conv. complete

Read result



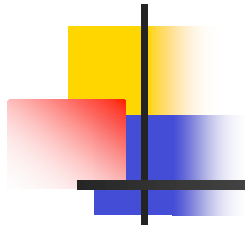
# A/D Conversion with the ATD10B8C

---

Where to look:

- Chapter 8 in Reference Manual
- Pg. 223-249 in Reference Guide





# References

---

- Ron Bishop, “Basic Microprocessors and the 6800”, Hayden Book Company Inc., 1979
- Motorola, “MC912SC Family Data Sheet”, Motorola, Inc., Rev. 5, 2007.
- Motorola, “MC912SC Reference Manual”, Motorola, Inc., Rev. 4, 2007.
- Motorola, “MC912SC Programming Reference Guide”, Motorola, Inc., Rev. 2, 2007.
- “Analog to Digital Converters”,  
[http://en.wikipedia.org/wiki/Analog-to-digital\\_converter](http://en.wikipedia.org/wiki/Analog-to-digital_converter)
- Thomas E. Brewer, “Experiments in Analog & Digital Electronics”, Kendall/Hunt Publishing Company, 2006.
- Dr. Ume, [http://www.me.gatech.edu/mechatronics\\_course/](http://www.me.gatech.edu/mechatronics_course/)