NUS
National University
of Singapore

Analog Interfacing

Ravi Suppiah
Lecturer, NUS SoC

Why It's Needed

- Embedded systems often need to measure values of physical parameters
- These parameters are usually continuous (*analog*) and not in a digital form which computers (which operate on discrete data values) can process

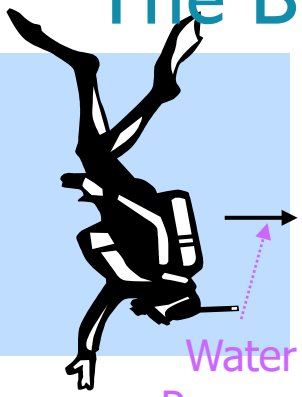
- Temperature
  - Thermometer (do you have a fever?)
  - Thermostat for building, fridge, freezer
  - Car engine controller
  - Chemical reaction monitor
  - Safety (e.g. microprocessor processor thermal management)
- Light (or infrared or ultraviolet) intensity
  - Digital camera
  - IR remote control receiver
  - Tanning bed
  - UV monitor
- Rotary position
  - Wind gauge
  - Knobs
- Pressure
  - Blood pressure monitor
  - Altimeter
  - Car engine controller
  - Scuba dive computer
  - Tsunami detector
- Acceleration
  - Air bag controller
  - Vehicle stability
  - Video game remote
- Mechanical strain
- Other
  - Touch screen controller
  - EKG, EEG
  - Breathalyzer

NUS
National University
of Singapore

CONVERTING BETWEEN ANALOG AND DIGITAL VALUES

ARM

The Big Picture – A Depth Gauge



Water Pressure

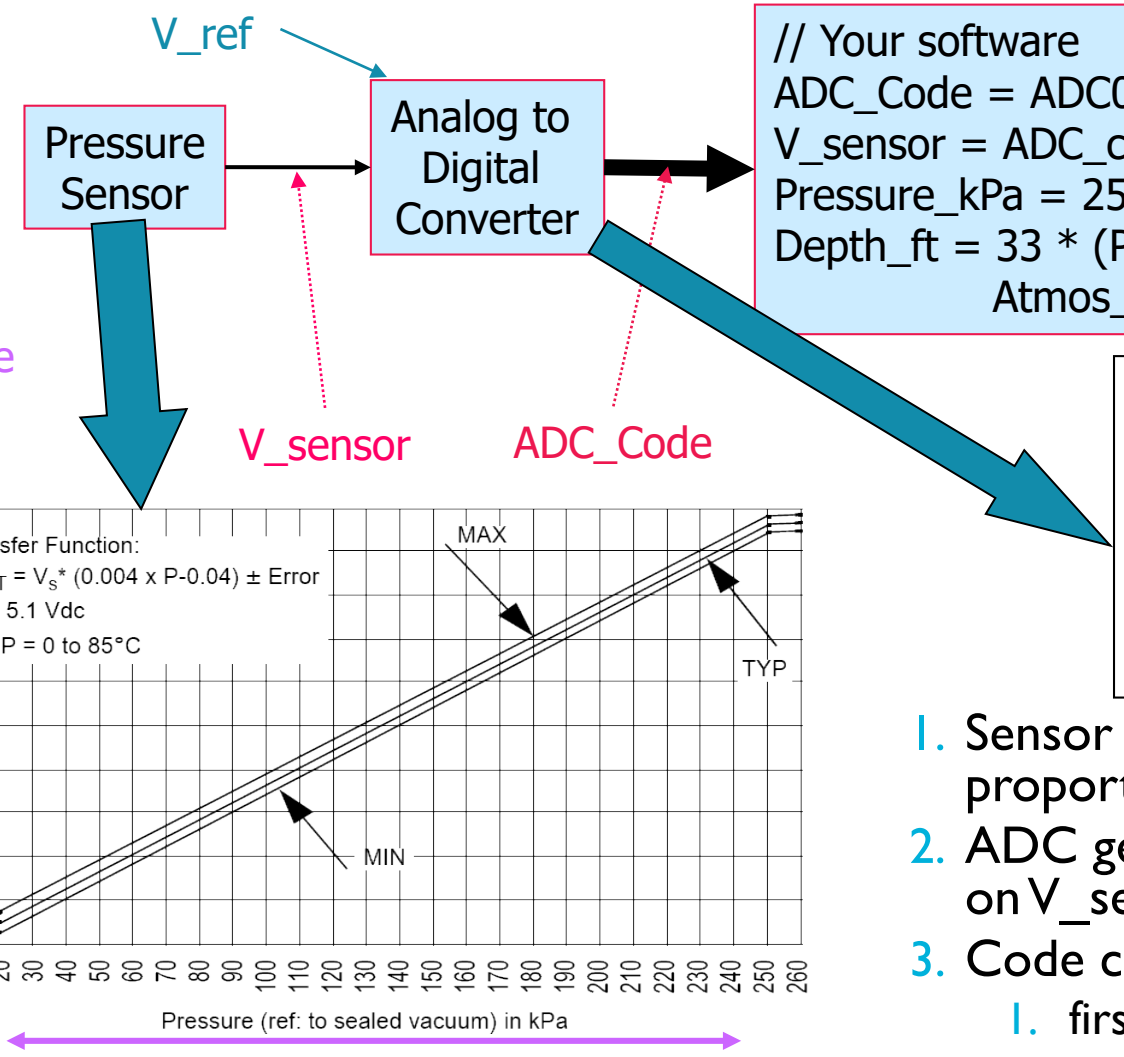
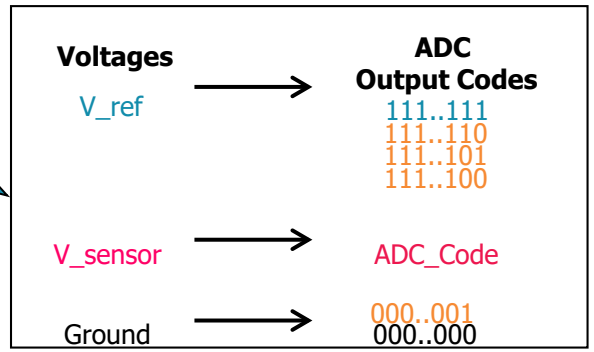


Figure 4. Output vs. Absolute Pressure

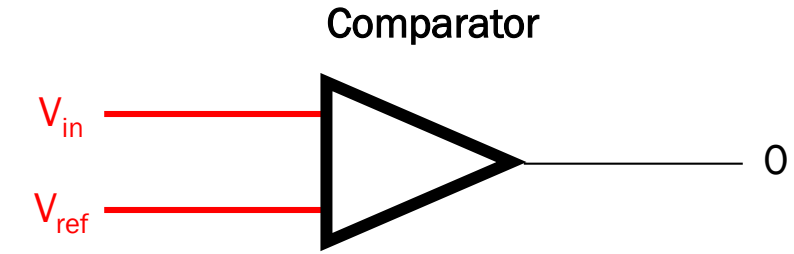
```
// Your software
ADC_Code = ADC0->R[0];
V_sensor = ADC_code*V_ref/1023;
Pressure_kPa = 250 * (V_sensor/V_supply+0.04);
Depth_ft = 33 * (Pressure_kPa –
                 Atmos_Press_kPa)/101.3;
```



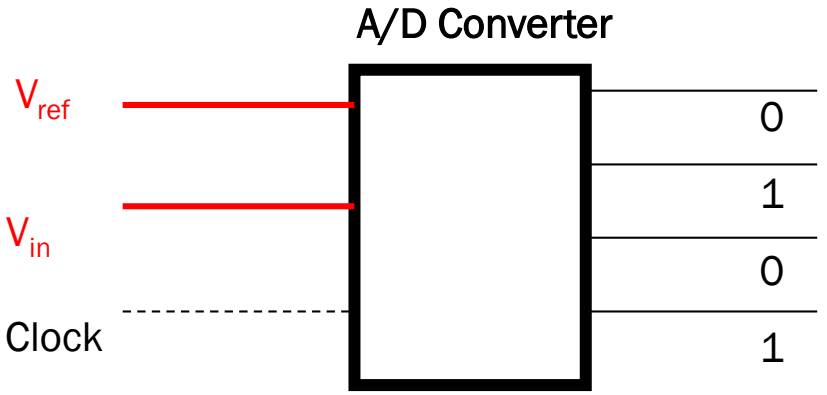
1. Sensor detects *water pressure* and generates a proportional *output voltage* V_sensor
2. ADC generates a proportional digital *integer* (code) based on V_sensor and V_ref
3. Code can convert that integer to a something more useful
   1. first a float representing the *voltage*,
   2. then another float representing *pressure*,
   3. finally another float representing *depth*

Getting From Analog to Digital

- A **Comparator** tells us "Is $V_{in} > V_{ref}$?"
  - Compares an **analog input voltage** with an **analog reference voltage** and determines which is larger, returning a 1-bit number
  - E.g. Indicate if depth > 100 ft
  - Set V_{ref} to voltage pressure sensor returns with 100 ft depth.

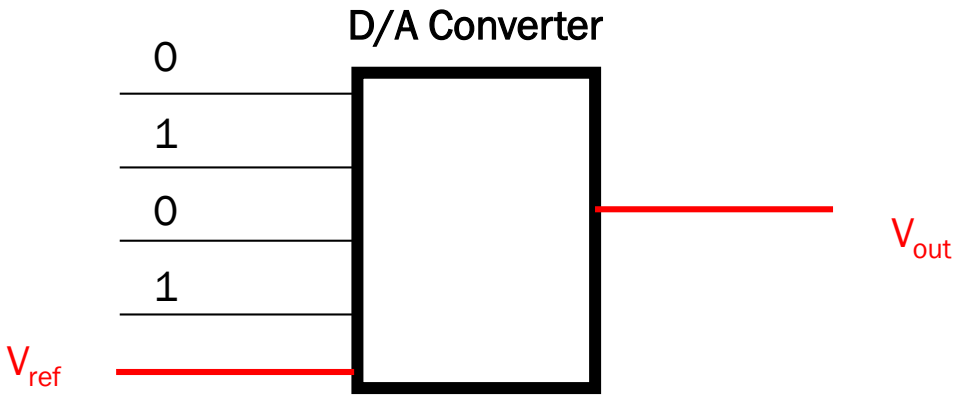


- An **Analog to Digital converter** [AD or ADC] tells us how large V_{in} is as a fraction of V_{ref}.
  - Reads an analog input signal (usually a voltage) and produces a corresponding multi-bit number at the output.
  - E.g. calculate the depth

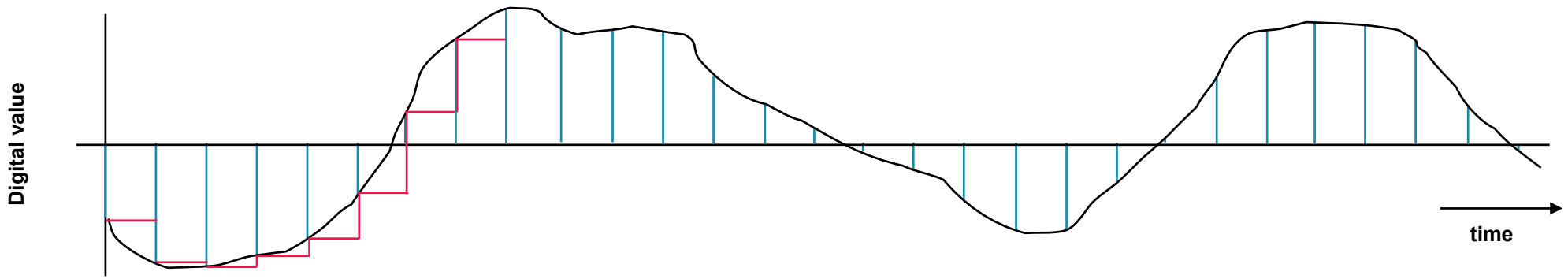


Digital to Analog Conversion

- May need to generate an analog voltage or current as an output signal
  - E.g. audio signal, video signal brightness.
- DAC: "Generate the analog voltage which is this fraction of V_{ref}"
- Digital to Analog Converter equation
  - *n* = input code
  - *N* = number of bits of resolution of converter
  - V_{ref} = reference voltage
  - V_{out} = output voltage. Either
    - $V_{out} = V_{ref} * n/(2^N)$ *or*
    - $V_{out} = V_{ref} * (n+1)/(2^N)$
    - *The offset +1 term depends on the internal tap configuration of the DAC – check the datasheet to be sure*



Waveform Sampling and Quantization



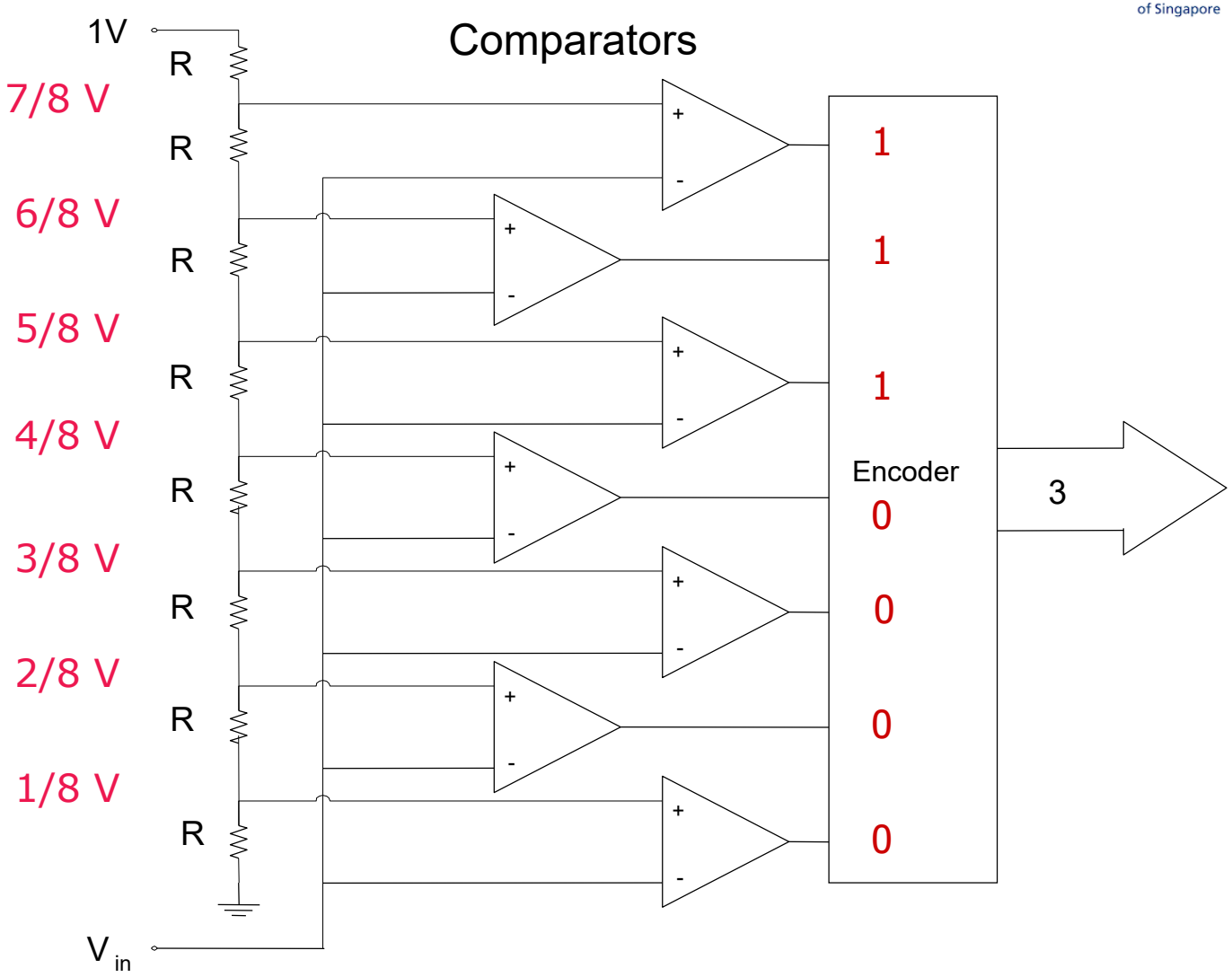
- A waveform is **sampled** at a constant rate – every Δt
  - Each such sample represents the instantaneous amplitude at the instant of sampling
  - *"At 37 ms, the input is 1.913419145134514512343 11… V"*
  - Sampling converts a **continuous time** signal to a **discrete time** signal

- The sample can now be **quantized** (converted) into a digital value
  - Quantization represents a **continuous** (analog) value with the closest **discrete** (digital) value
  - *"The sampled input voltage of 1.913419145134514512343 11… V is best represented by the code 0x018, since it is in the range of 1.901 to 1.9980 V which corresponds to code 0x018."*

ANALOG TO DIGITAL CONVERSION CONCEPTS

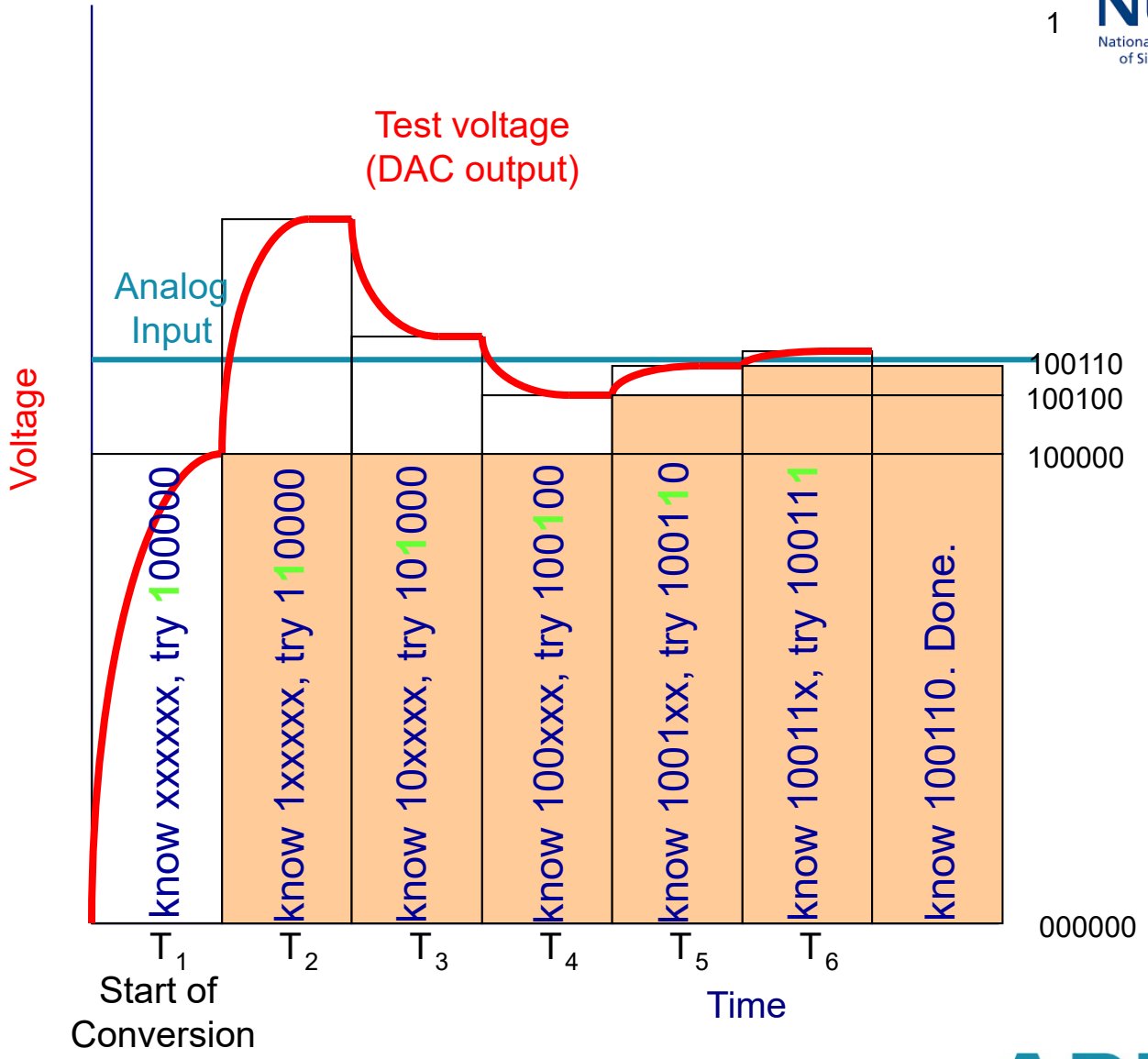
A/D – Flash Conversion

- A multi-level voltage divider is used to set voltage levels over the complete range of conversion.
- A comparator is used at each level to determine whether the voltage is lower or higher than the level.
- The series of comparator outputs are encoded to a binary number in digital logic (a priority encoder)
- Components used
  - 2^N resistors
  - 2^N-1 comparators
- Note
  - This particular resistor divider generates voltages which are *not* offset by ½ bit, so maximum error is 1 bit
  - We could change this offset voltage by using resistors of values R, 2R, 2R ... 2R, 3R (starting at bottom)



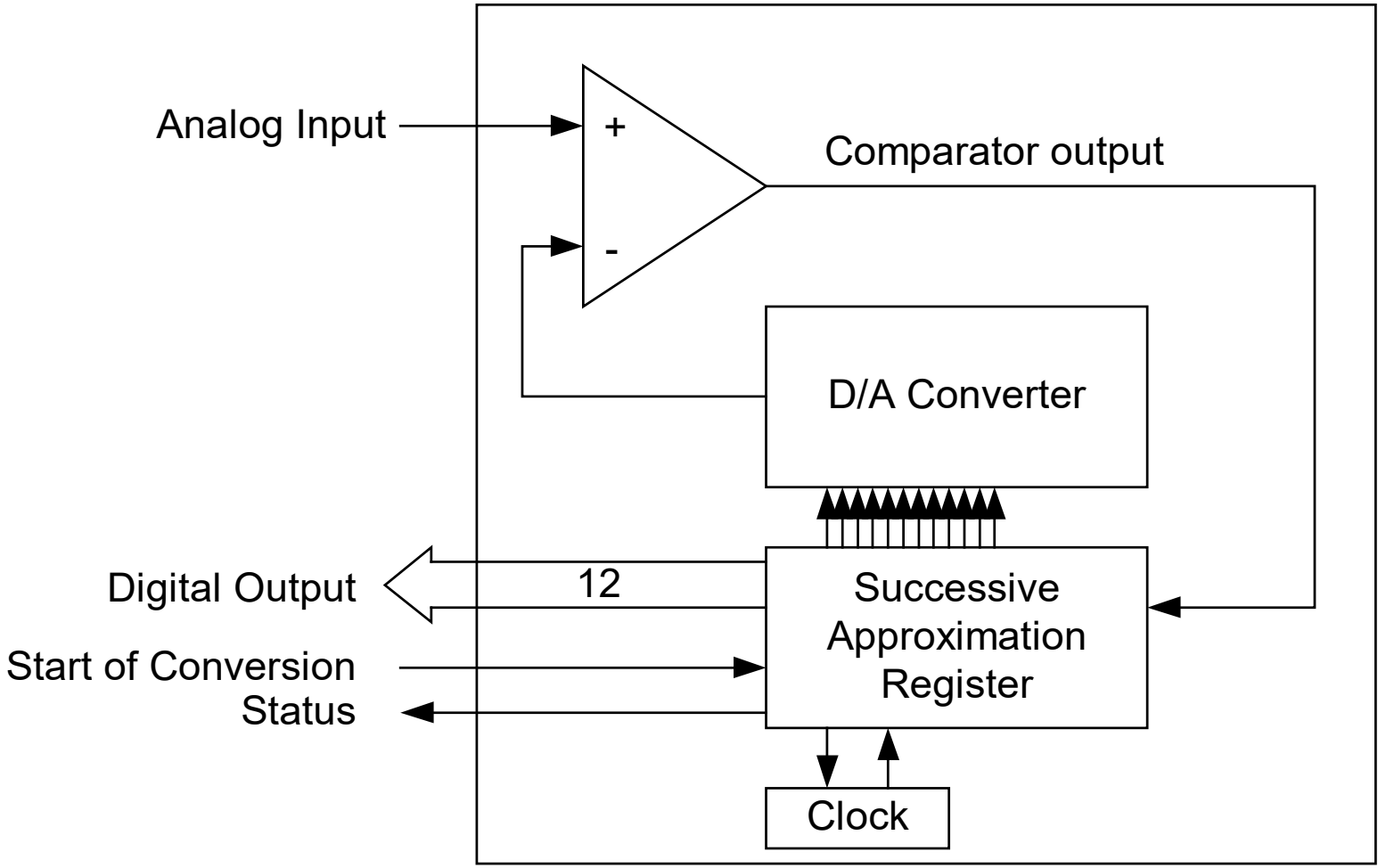
ADC - Successive Approximation Conversion

- Successively approximate input voltage by using a binary search and a DAC
- SA Register holds current approximation of result
- Set all DAC input bits to 0
- Start with DAC's most significant bit
- Repeat
  - Set next input bit for DAC to 1
  - Wait for DAC and comparator to stabilize
  - If the DAC output (test voltage) is **smaller** than the input then set the current bit to 1, else clear the current bit to 0



A/D - Successive Approximation

Converter Schematic



ADC Performance Metrics

- Linearity measures how well the transition voltages lie on a straight line.
- Differential linearity measure the equality of the step size.
- Conversion time: between start of conversion and generation of result
- Conversion *rate* = inverse of conversion *time*

Sampling Problems

NUS National University of Singapore

- Nyquist criterion
  - $F_{sample} >= 2 * F_{max\ frequency\ component}$
  - Frequency components above ½ F_{sample} are aliased, distort measured signal

- Nyquist and the real world
  - This theorem assumes we have a perfect filter with "brick wall" roll-off
  - Real world filters have more gentle roll-off
  - Inexpensive filters are even worse (e.g. first order filter is 20 dB/decade, aka 6 dB/octave)
  - So we have to choose a sampling frequency high enough that our filter attenuates aliasing components adequately

ARM

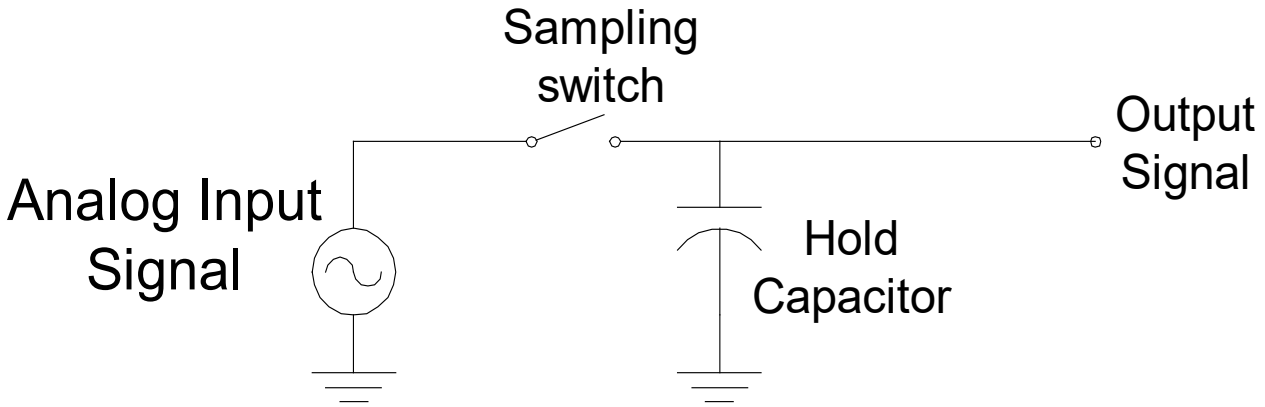
Inputs

NUS National University of Singapore

- Differential
  - Use two channels, and compute difference between them
  - Very good noise immunity
  - Some sensors offer differential outputs (e.g. Wheatstone Bridge)
- Multiplexing
  - Typically share a single ADC among multiple inputs
  - Need to select an input, allow time to settle before sampling
- Signal Conditioning
  - Amplify and filter input signal
  - Protect against out-of-range inputs with clamping diodes

ARM

Sample and Hold Devices



- Some A/D converters require the input analog signal to be held constant during conversion (e.g. successive approximation devices)
- In other cases, peak capture or sampling at a specific point in time requires a sampling device.
- A "sample and hold" circuit performs this operation
- Many A/D converters include a sample and hold circuit

NUS
National University
of Singapore

KL25 ANALOG INTERFACING PERIPHERALS

ARM

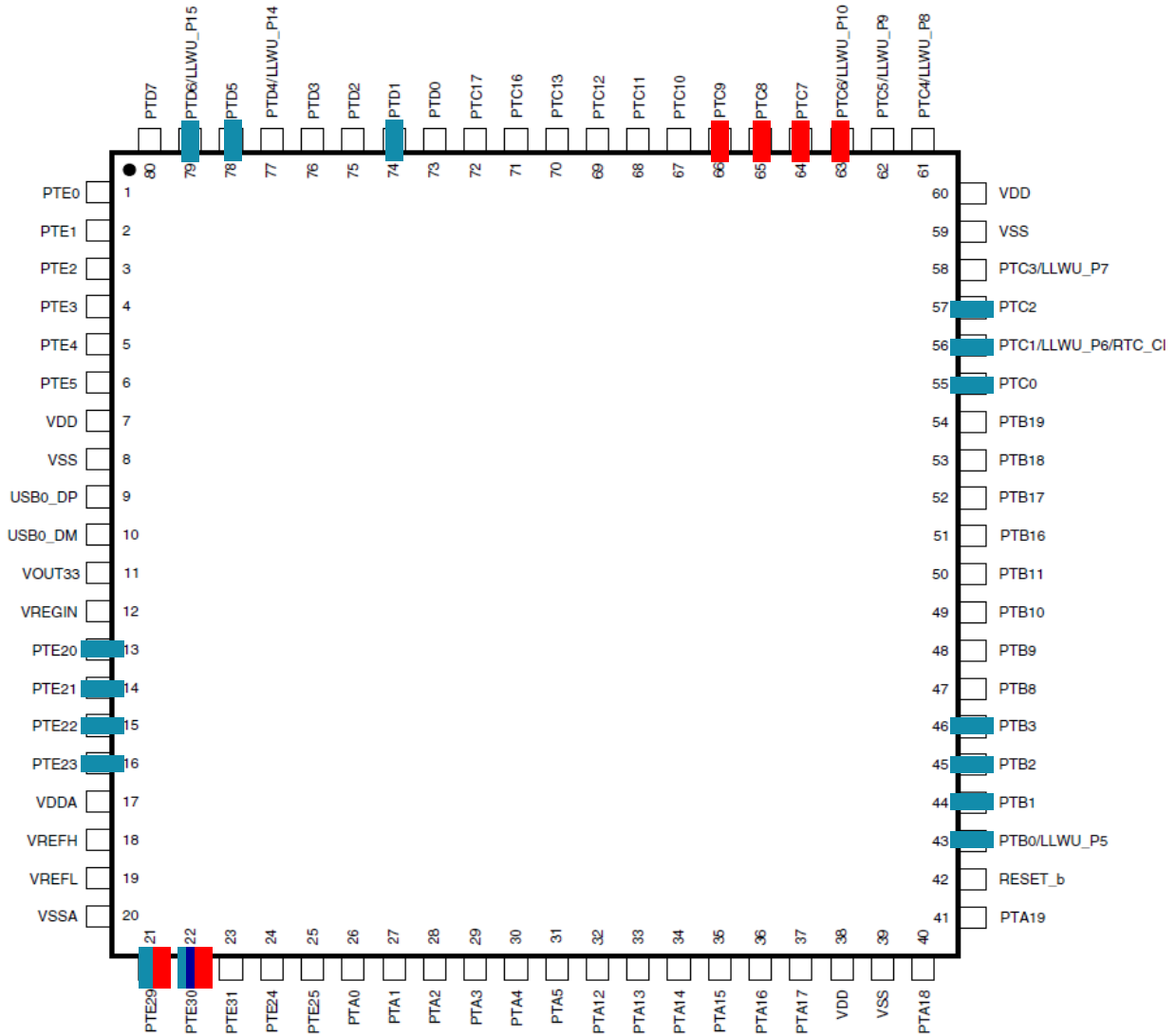
Sources of Information

- KL25 Subfamily Reference Manual (Rev. 1, June 2012)
  - Describes architecture of peripherals and their control registers
  - Digital to Analog Converter
    - Chapter 30 of KL25 Subfamily Reference Manual
  - Analog Comparator
    - Chapter 29 of KL25 Subfamily Reference Manual
  - Analog to Digital Converter
    - Chapter 28 of KL25 Subfamily Reference Manual

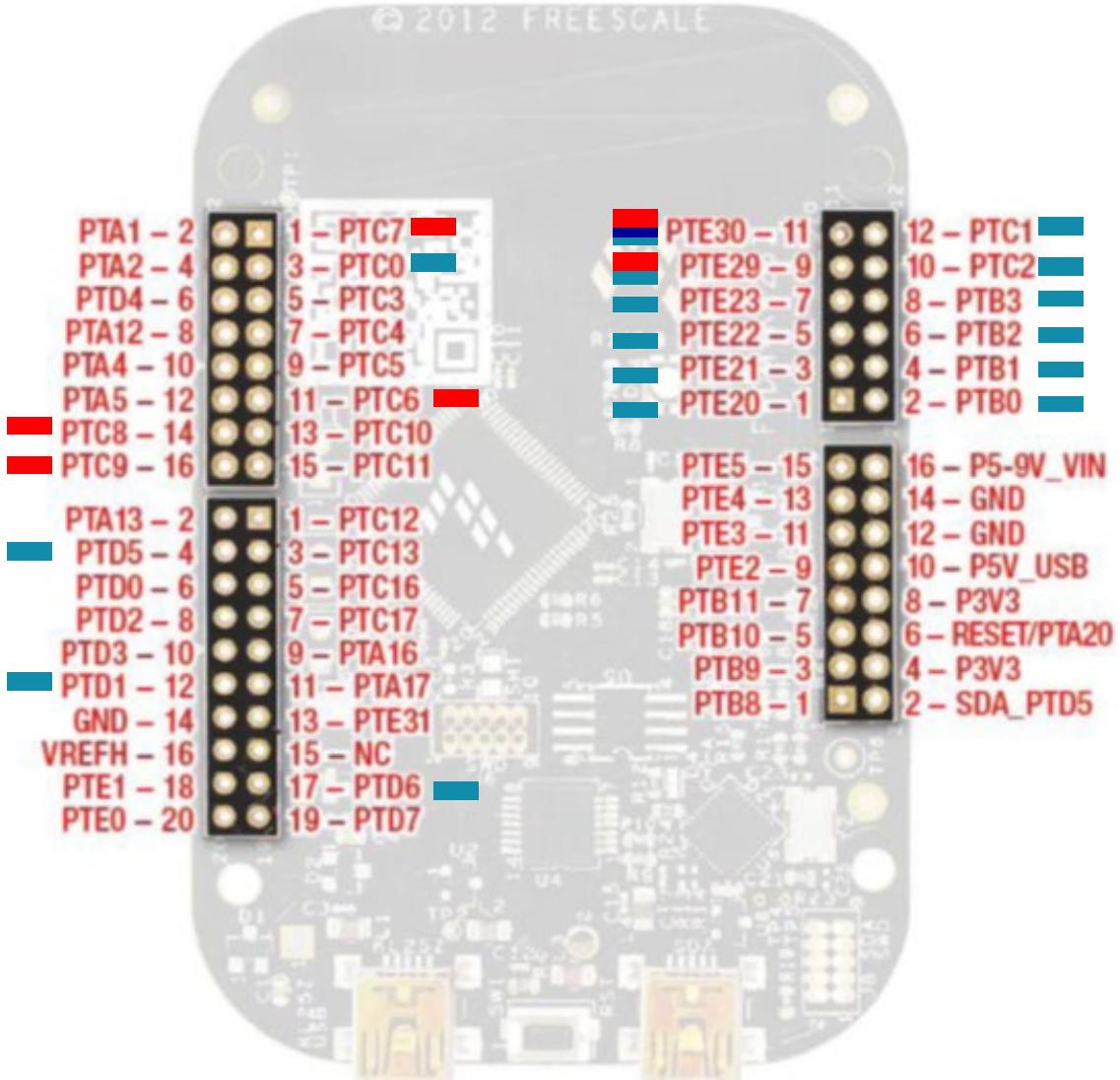
- KL25 Sub-family Data Sheet (Rev. 3, 9/19/2012)
  - Describes circuit-specific performance parameters: operating voltages, min/max speeds, cycle times, delays, power and energy use

KL25Z Analog Interface Pins

- 80-pin QFP
- Inputs
  - 1 16-bit ADC with 14 input channels
  - 1 comparator with 6 external inputs, one 6-bit DAC
- Output
  - 1 12-bit DAC



Freedom KL25Z Analog I/O



Inputs

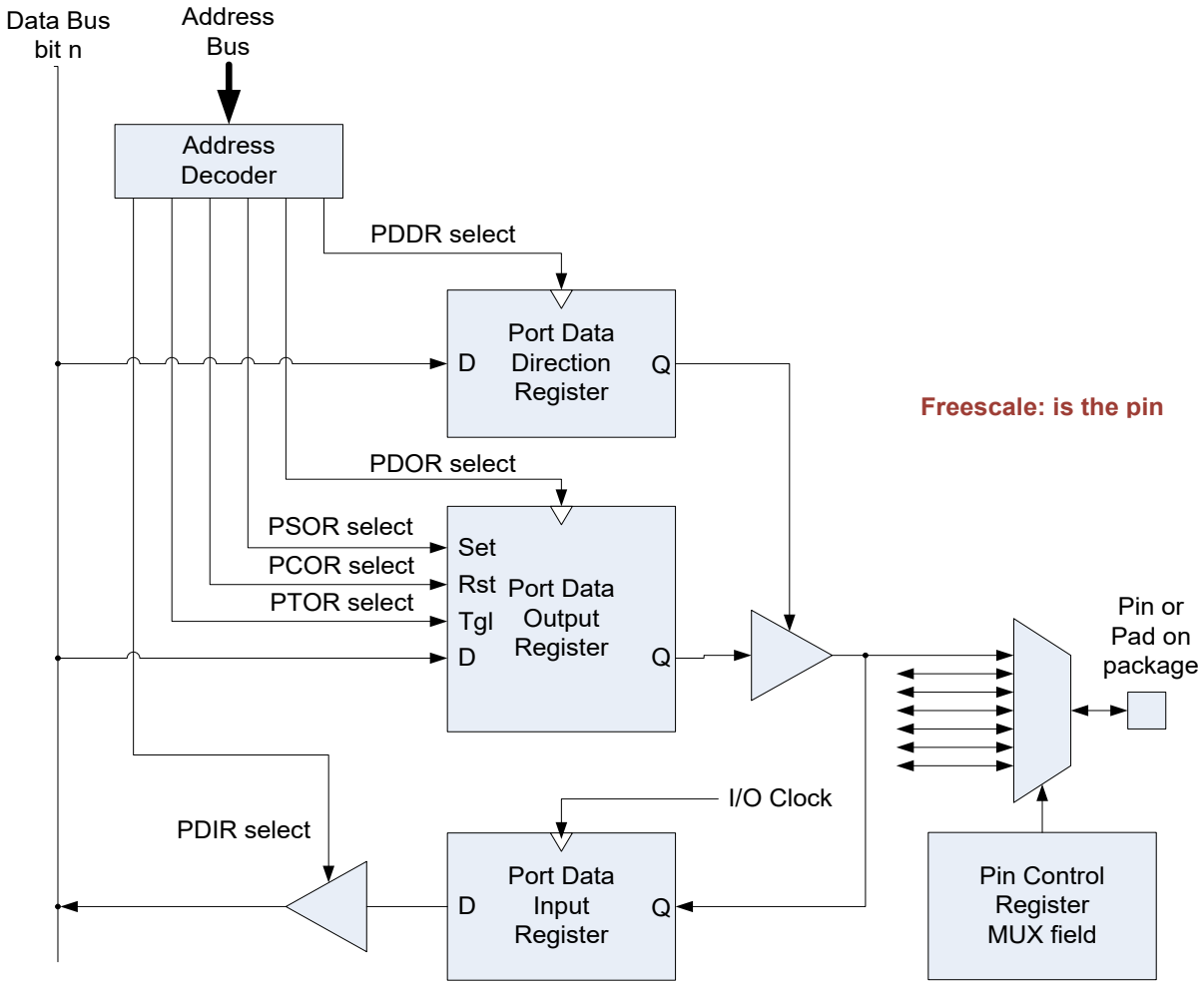
- 14 external ADC channels
- 6 external comparator channels

Output

- 1 12-bit DAC

Using a Pin for Analog Input or Output

- Configuration
  - Direction
  - MUX
- Data
  - Output (different ways to access it)
  - Input



Freescale: is the pin

Pin Control Register to Select MUX Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					MUX			0	DSE	0	PFE	0	SRE	PE	PS
W																
Reset	0	0	0	0	0	x*	x*	x*	0	x*	0	x*	0	x*	x*	x*

80 LQFP	64 LQFP	48 QFN	32 QFN	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
64	52	40	28	PTC7	CMP0_IN1	CMP0_IN1	PTC7	SPI0_MISO			SPI0_MOSI		
65	53	—	—	PTC8	CMP0_IN2	CMP0_IN2	PTC8	I2C0_SCL	TPM0_CH4				

- MUX field of PCR defines connections

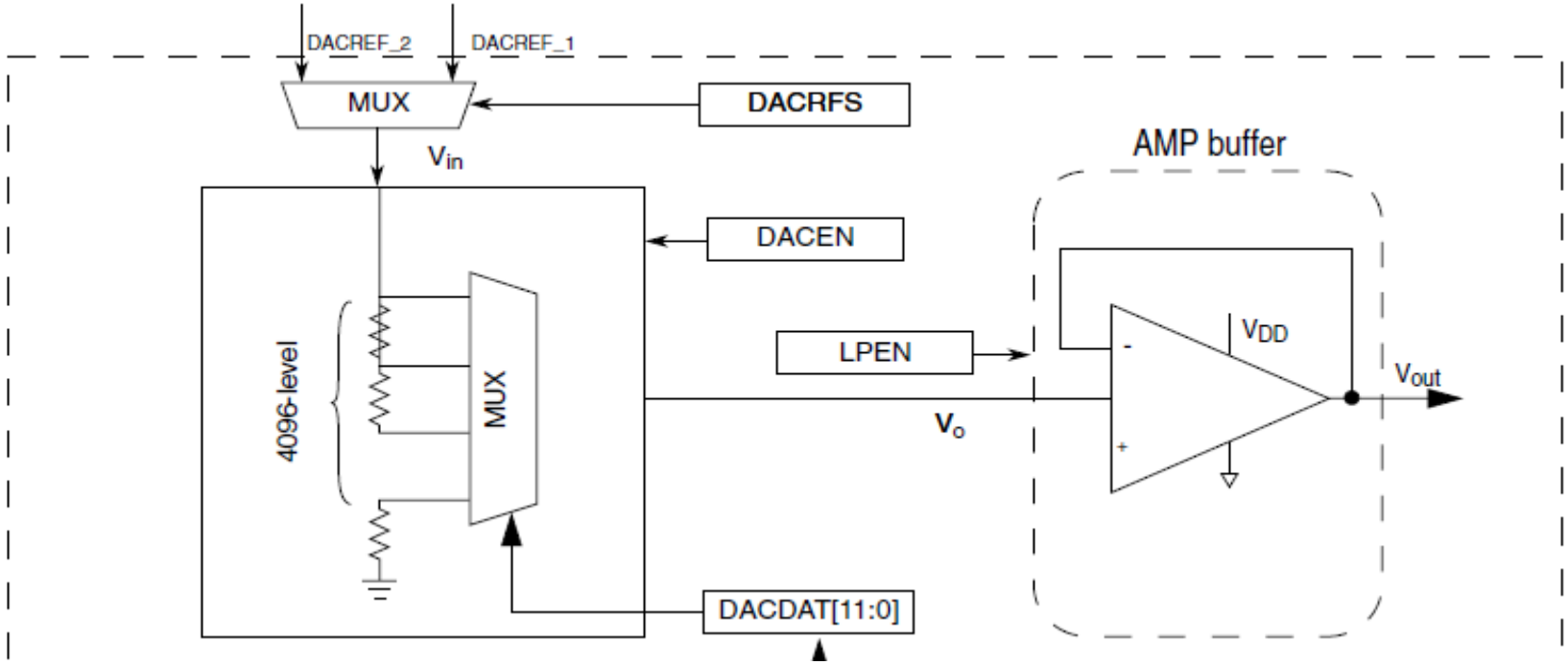
MUX (bits 10-8)	Configuration
000	Digital circuits disabled, analog enabled
001	Alternative 1 – GPIO
010	Alternative 2
011	Alternative 3
100	Alternative 4
101	Alternative 5
110	Alternative 6
111	Alternative 7

```
PORTC->PCR[7] &= ~PORT_PCR_MUX_MASK;
PORTC->PCR[7] |= PORT_PCR_MUX(0);
```

DIGITAL TO ANALOG CONVERTER

ARM

DAC Overview



- Load DACDAT with 12-bit data N
- MUX selects a node from resistor divider network to create $V_o = (N+1)*V_{in}/2^{12}$
- V_o is buffered by output amplifier to create V_{out}
  - $V_o = V_{out}$ but V_o is high impedance - can't drive much of a load, so need to buffer it

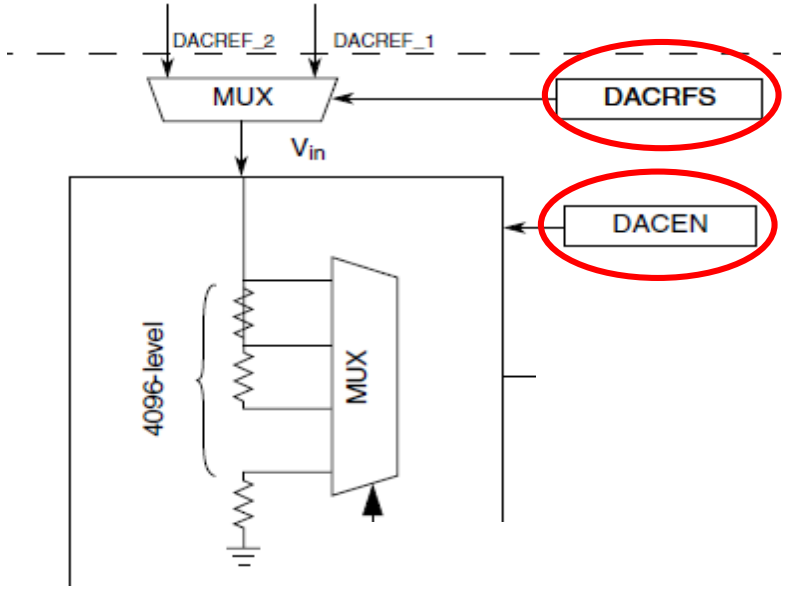
DAC Operating Modes

- Normal
  - DAT0 is converted to voltage immediately
- Buffered
  - Data to output is stored in 16-word buffer
  - Next data item is sent to DAC when a selectable trigger event occurs
    - Software Trigger - write to DACSWTRG field in DACx_C0
    - Hardware Trigger - from PIT timer peripheral
  - Normal Mode
    - Circular buffer
  - One-time Scan Mode
    - Pointer advances until reaching upper limit of buffer, then stops
  - Status flags in DACx_SR

DAC Control Register 0: DACx_C0

Bit	7	6	5	4	3	2	1	0
Read	DACEN	DACRFS	DACTRGSEL	0	LPEN	0	DACBTIEN	DACBBIEN
Write				DACSWTRG				
Reset	0	0	0	0	0	0	0	0

- DACEN - DAC Enabled when 1
- DACRFS - DAC reference voltage select
  - 0: DACREF_1. Connected to VREFH
  - 1: DACREF_2. Connected to VDDA
- LPEN - low-power mode
  - 0: High-speed mode. Fast (15 us settling time) but uses more power (up to 900 uA supply current)
  - 1: Low-power mode. Slow (100 us settling time) but more power-efficient (up to 250 uA supply current)
- Additional control registers used for buffered mode



DAC Control Register 1: DACx_C1

Bit	7	6	5	4	3	2	1	0
Read	DMAEN	0				DACBFMD	0	DACBFEN
Write								
Reset	0	0	0	0	0	0	0	0

- DACBFEN
  - 0: Disable buffer mode
  - 1: Enable buffer mode
- DACBFMD - Buffer mode select
  - 0: Normal mode (circular buffer)
  - 1: One-time scan mode

DAC Data Registers

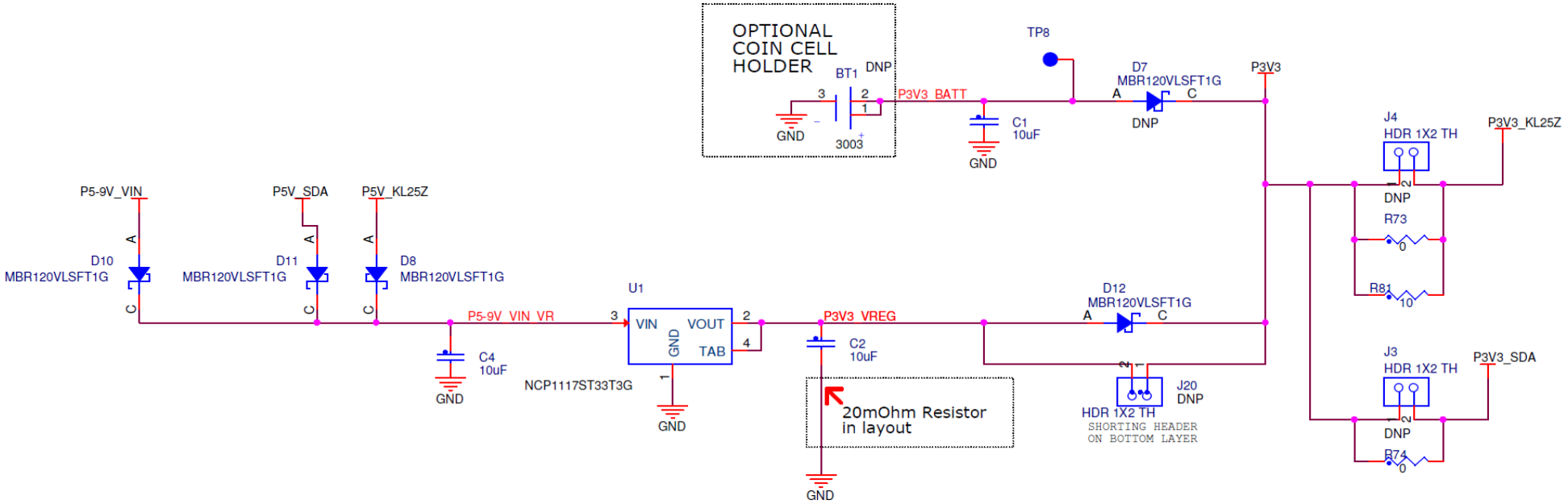
- These registers are only eight bits long
- DATA[11:0] stored in two registers
  - DATA0: Low byte [7:0] in DACx_DATnL
  - DATA1: High nibble [11:0] in DACx_DATnH

Example: Waveform Generator

- Supply clock to DAC0 module
  - Bit 31 of SIM SCGC6
- Set Pin Mux to Analog (0)
- Enable DAC
- Configure DAC
  - Reference voltage
  - Low power mode?
  - Normal mode (not buffered)
- Write to DAC data register

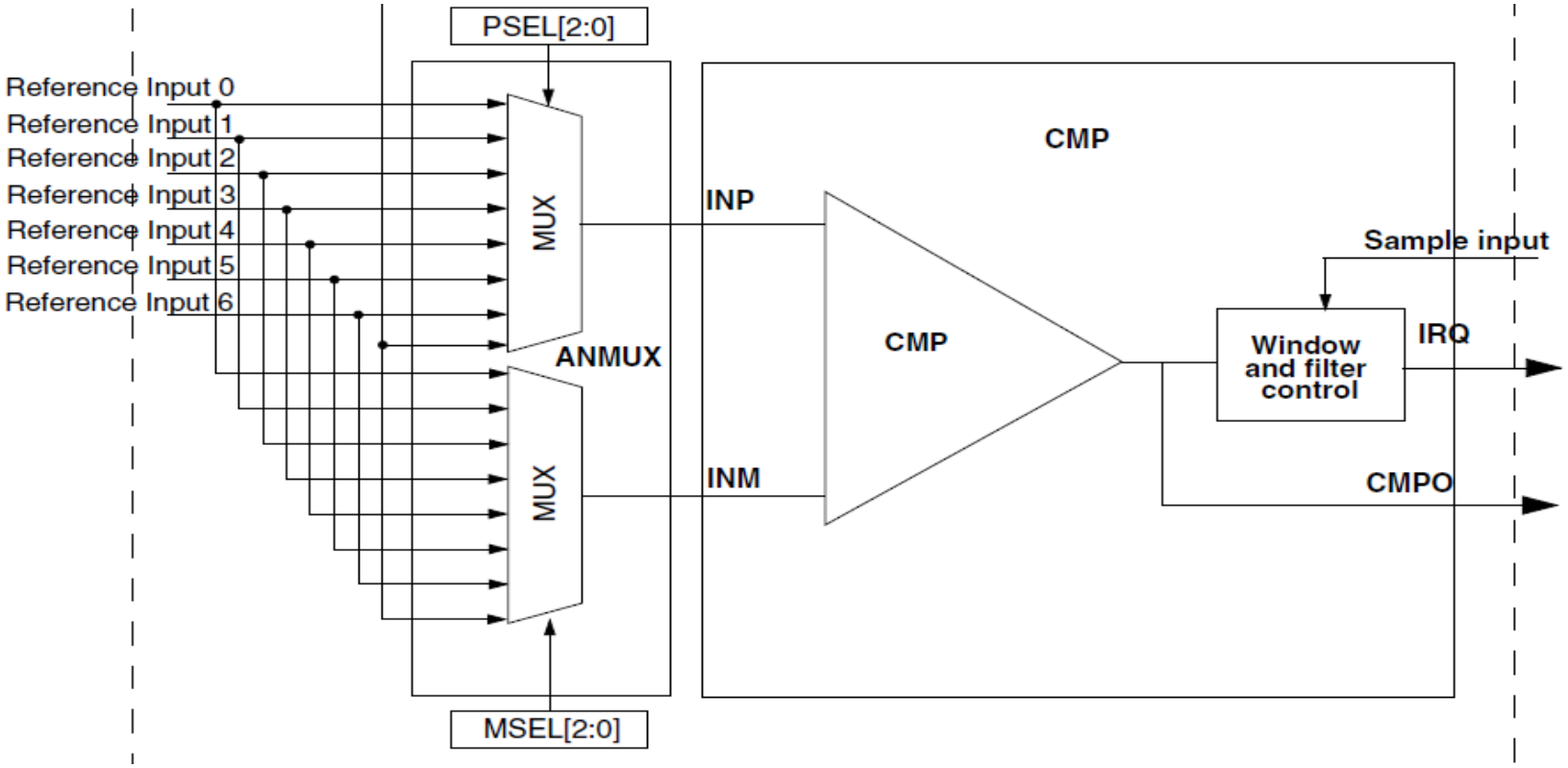
ANALOG COMPARATOR

Example: Power Failure Detection



- Need warning of when power has failed
  - Very limited amount of time before capacitor C2 discharges
  - Save critical information
  - Turn off output devices
  - Put system into safe mode
- Can use a comparator to compare V_{in} against a fixed reference voltage V_{Ref}

Comparator Overview



- Comparator compares INP and INM
- CMPO Output indicates if INP>INM (1) or INP<INM (0)
- Can generate an interrupt request (+, -, or +- edges)
- ANMUX selection of one of multiple reference inputs, using PSEL and MSEL fields

The End!

NUS National University of Singapore

- We are done with Analog! For Now…

ARM