Analogue to Digital Conversion II

ENCE361 Embedded Systems 1

Course Coordinator: Ciaran Moore (ciaran.moore@Canterbury.ac.nz)

Lecturer: Le Yang (le.yang@canterbury.ac.nz)

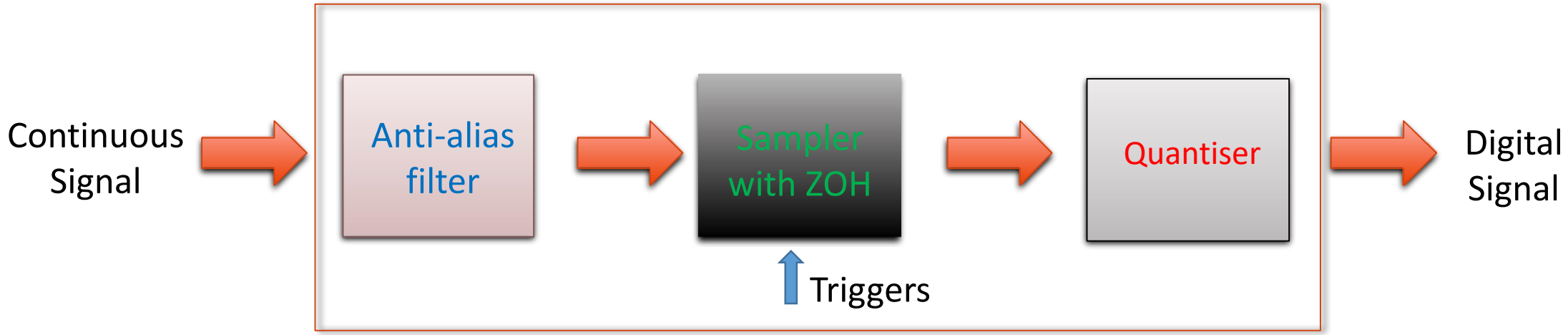
Department of Electrical and Computer Engineering

Where we're going today

- **Uniform quantisation**
- Flash and sigma-delta quantisers
- Dual-slop integrating and successive-approximation quantisers
- ADC in Tiva C-series Launchpad
- Homework

Analogue to Digital Conversion Review

- ADC diagram



- Anti-alias filter removes frequencies of the input that might alias
- Sampler with zero-order hold (ZOH) holds the input value until next conversion trigger
- Quantiser maps the held input value into **one of possible discrete values**

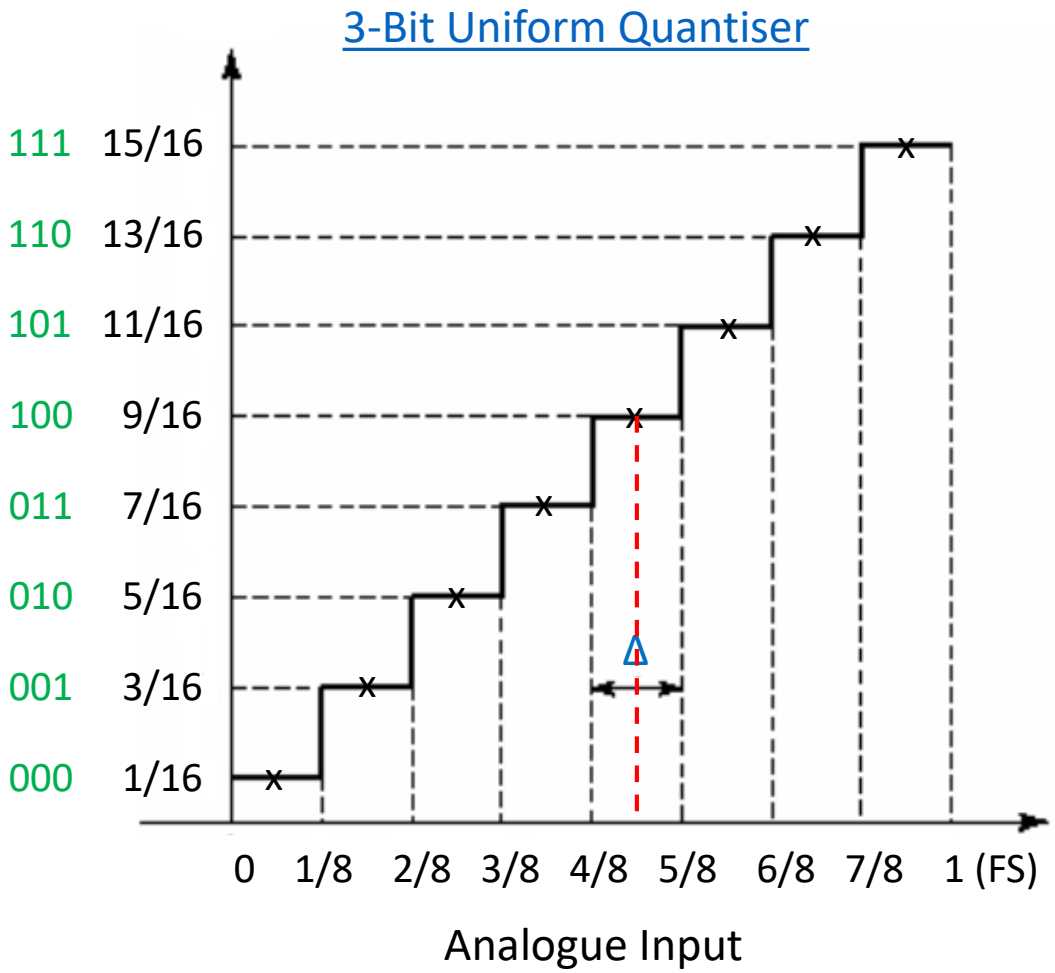
N-bit Uniform Quantisation

- N-bit uniform quantiser divides voltage range into 2^N equal intervals with quantization step

$$\Delta = (V_{max} - V_{min})/ 2^N$$

- For voltage in interval $[k\Delta, (k+1)\Delta]$, k=0,1,…, 2^N-1, output the binary representation of k
  - Discrete voltage = $(k+(k+1))\Delta/2$
  - Encode input voltage using N bits
  - Maximum quantisation error amplitude = $\Delta/2$

- Key design question for a uniform quantiser
  - How to determine which quantisation interval the given input belongs to (i.e., how to find k?)



Quantisation Error

$N=3 \quad 2^{-4} = \frac{1}{16} > 5\% \quad \frac{1}{20} = 5\%$

- Quantisation error is almost unavoidable in each ADC conversion
- For N-bit uniform quantisation, maximum error amplitude = $\Delta/2$

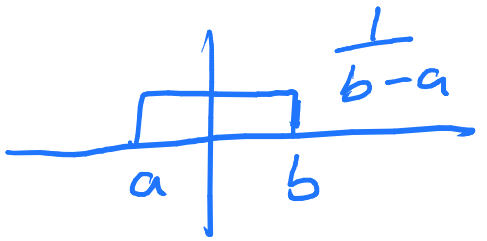
- Relative maximum amplitude error

$$\frac{\Delta/2}{(V_{max}-V_{min})} = \frac{\left(\frac{V_{max}-V_{min}}{2^N}\right)/2}{(V_{max}-V_{min})} = 2^{-(N+1)}$$

- Quantisation noise power = $\Delta^2/12$
- Exercise: for a sinewave of amplitude A, with N-bit uniform quantisation, what is the signal-to-noise ratio (SNR)? (more in the next lecture)

Where we're going today

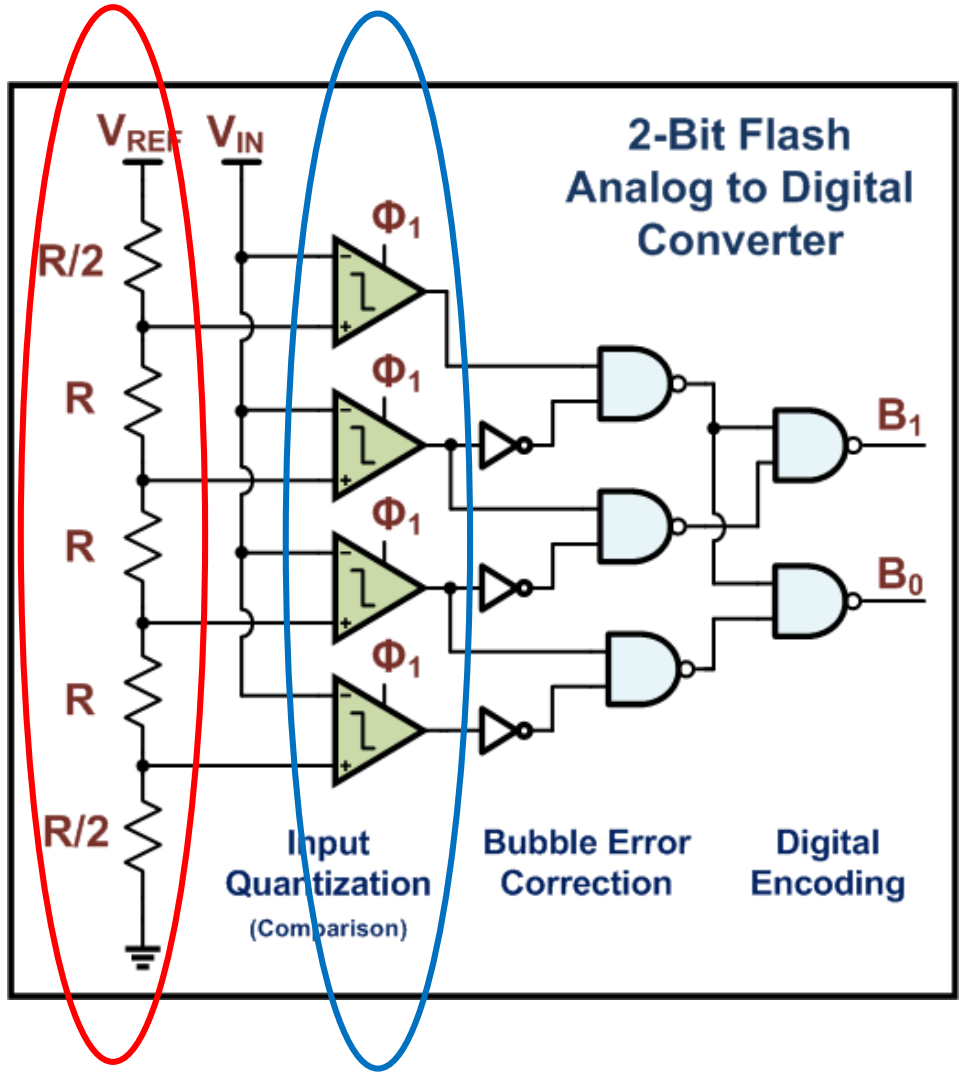
$x \sim U(a, b)$



- Uniform quantization
- **Flash and Sigma-Delta quantisers**
- Dual-slop integrating and successive-approximation quantisers
- ADC in Tiva C-series Launchpad
- Homework

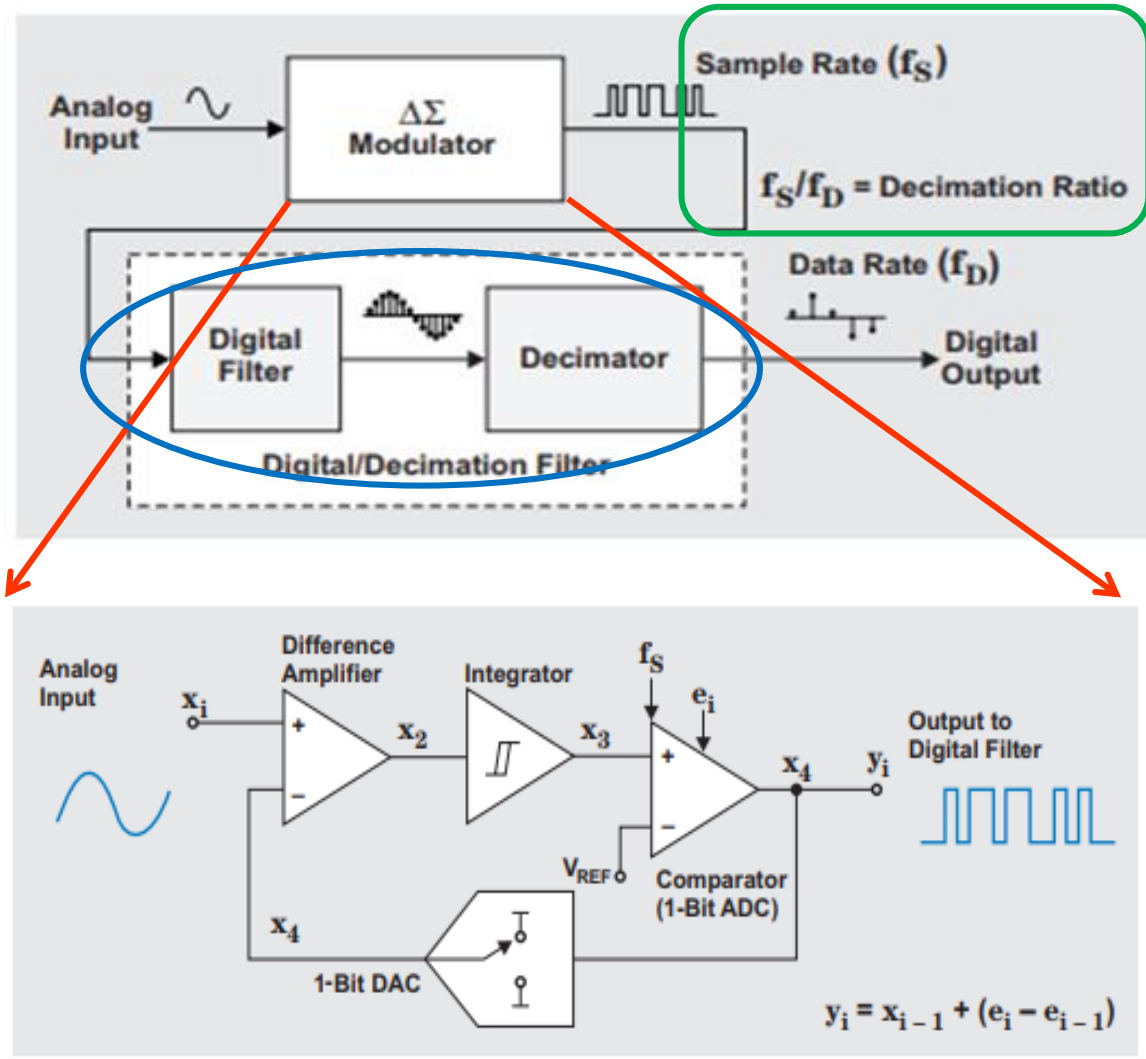
Flash Quantiser

- Use voltage ladder to generate successive reference voltages
  - Produce 2^N quantization intervals simultaneously
- Use 2^N comparators to find the quantisation interval the input V_{IN} belongs to, in a single step
- Fastest, simple but could require a huge number of comparators



Sigma-Delta Quantiser

- Oversample the analog input
- Sigma-Delta modulator
  - The number of ‘ones’ in the modulator output is proportional to input value
- Digital/decimator accumulates modulator output over time
  - Output digital signals with required sampling rate
- High resolution, popular in communications engineering



From http://www.ti.com/lit/an/slyt423a/slyt423a.pdf

Where we're going today

- Uniform quantization
- Flash and Sigma-Delta quantisers
- **Dual-slop integrating and successive-approximation quantisers**
- ADC in Tiva C-series Launchpad
- Homework

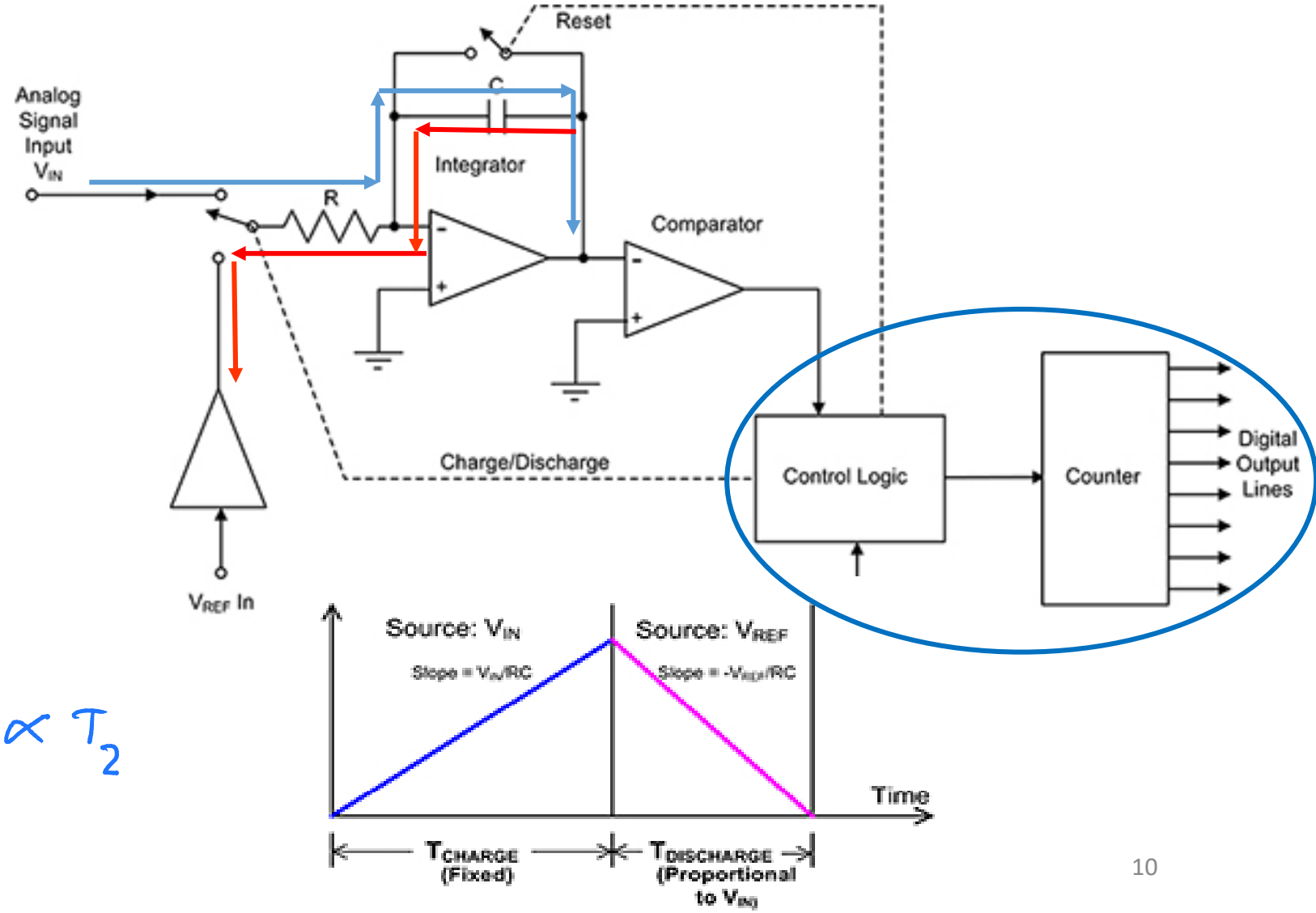
Dual-Slope Integrating Quantiser

- **Phase 1**: reset and charge C using analog input for fixed period T_1
- **Phase 2**: Discharge C using reference input until voltage reaches 0 (say after T_2)

$$V_{IN}T_1 = -V_{REF}T_2$$

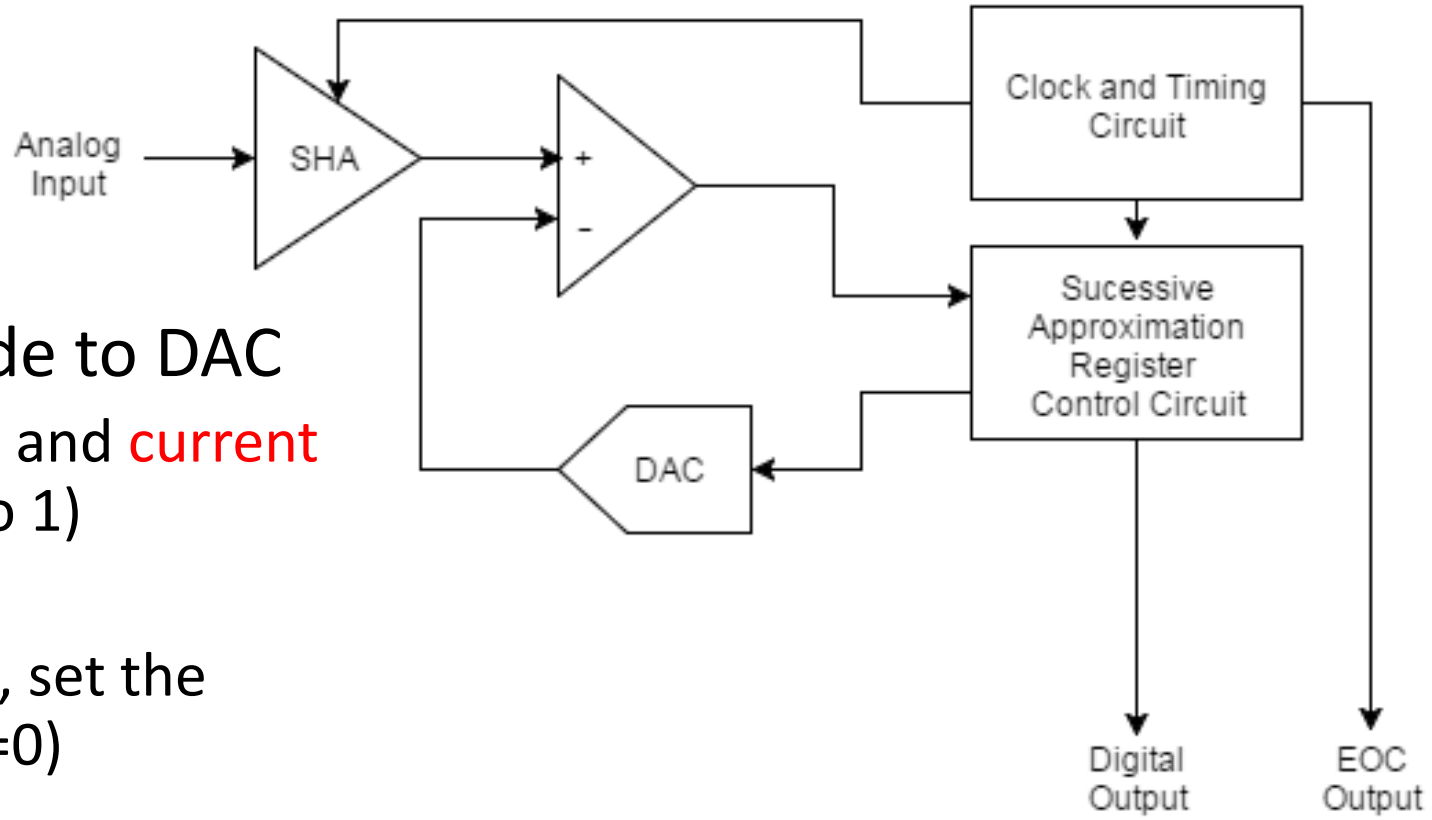
$$V_{IN} = -V_{REF}T_2/T_1$$

$V_{in} \propto T_2$



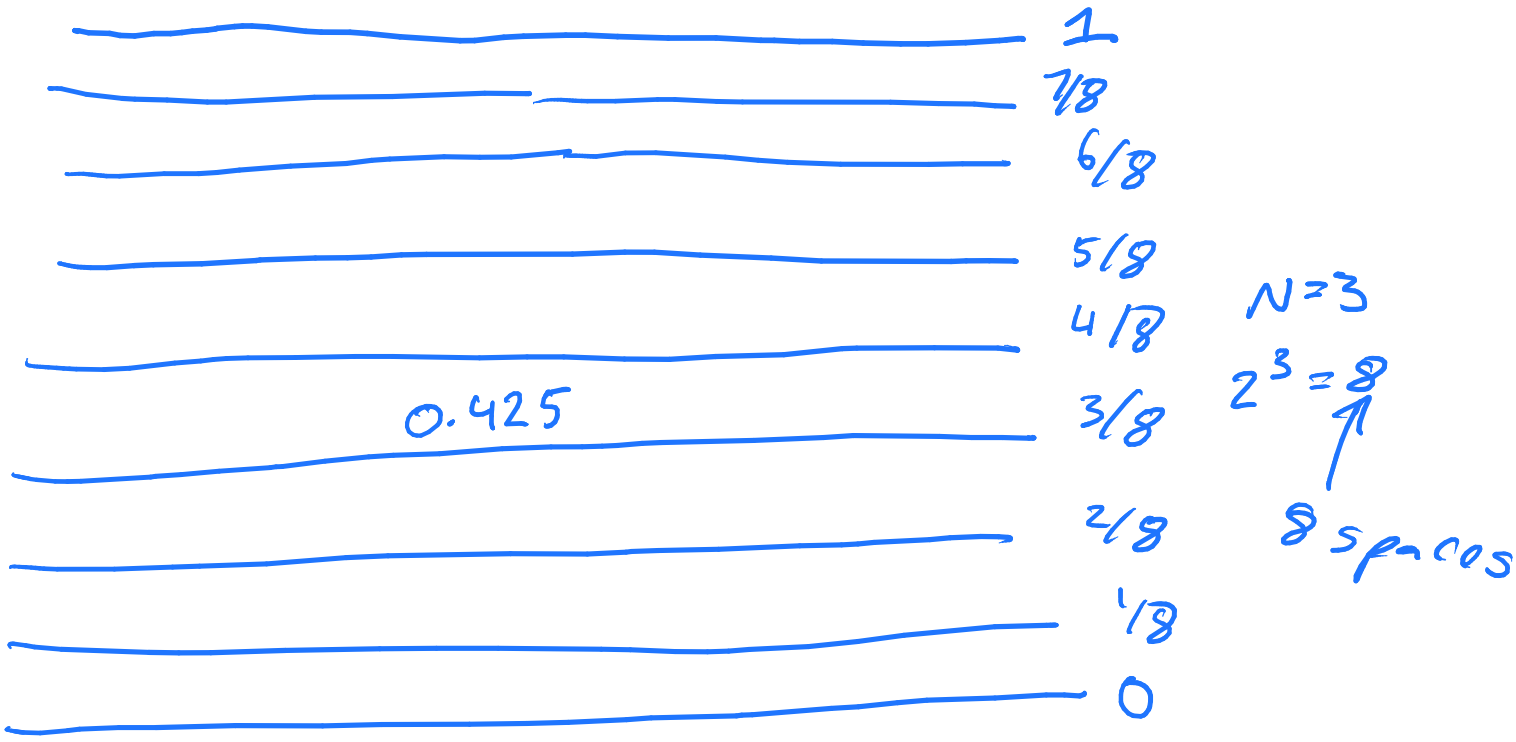
Successive-Approximation Quantiser (1)

- Hold the analog input



- Control unit sends binary code to DAC
  - Composed of determined bits and current bit under consideration (set to 1)
  - DAC produces a value
  - If analog Input >= DAC output, set the current bit (=1), else clear it (=0)
- Continue until all N bits are determined
  - At least N clock cycles are needed

From https://www.allaboutcircuits.com/technical-articles/understanding-analog-to-digital-converters-the-successive-approximation-reg/


1
7/8
6/8
5/8
4/8
0.425
3/8
2/8
1/8
0
N=3
$2^3=8$
8 spaces

Successive-Approximation Quantiser (2)

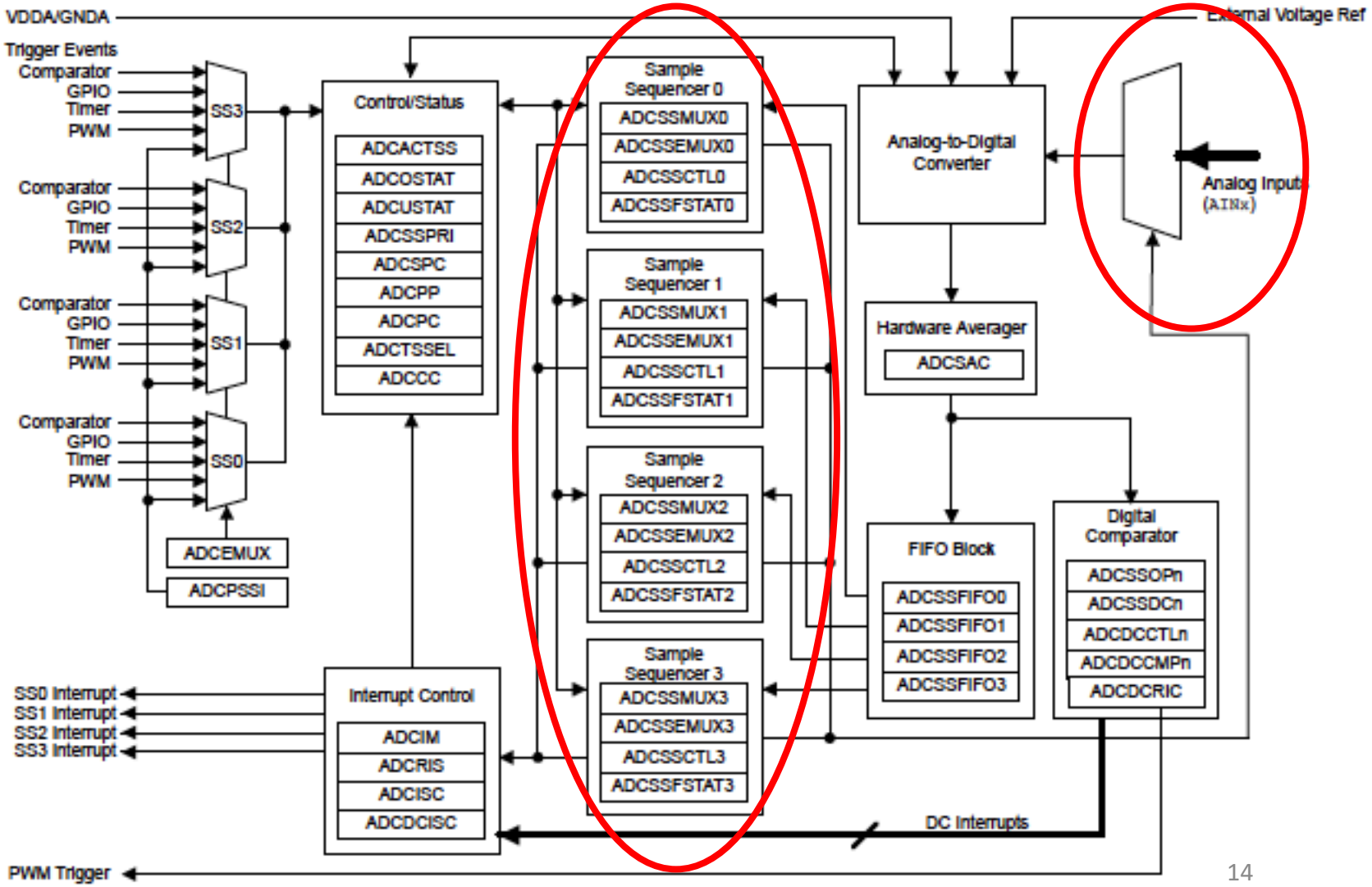
- Successive-approximation quantisation example
  - 3-bit uniform quantisation with V_{max} = 1 V and V_{min} = 0 V, quantization step Δ = 1/8
  - Analogue input = 0.425 V
  - Control unit starts with the binary code 100 $4 \times \frac{1}{8} = 0.5$
    - DAC outputs 0.5 V (= 4*Δ) > 0.425 V → the first bit = 0
  - Control unit proceeds with the binary code 010 $2 \times \frac{1}{8} = \frac{1}{4} = 0.25$
    - DAC outputs 0.25 V (= 2*Δ) <0.425 V → the second bit = 1
  - Control unit proceeds with the binary code 011 $3 \times \frac{1}{8} = \frac{3}{8}$
    - DAC outputs 0.375 V (= 3*Δ) <0.425 V → the third bit = 1
  - Control unit outputs with the quantisation result 011

Where we're going today

- Uniform quantization
- Flash and Sigma-Delta quantisers
- Dual-slop integrating and successive-approximation quantisers
- **ADC in Tiva C-series Launchpad**
- Homework

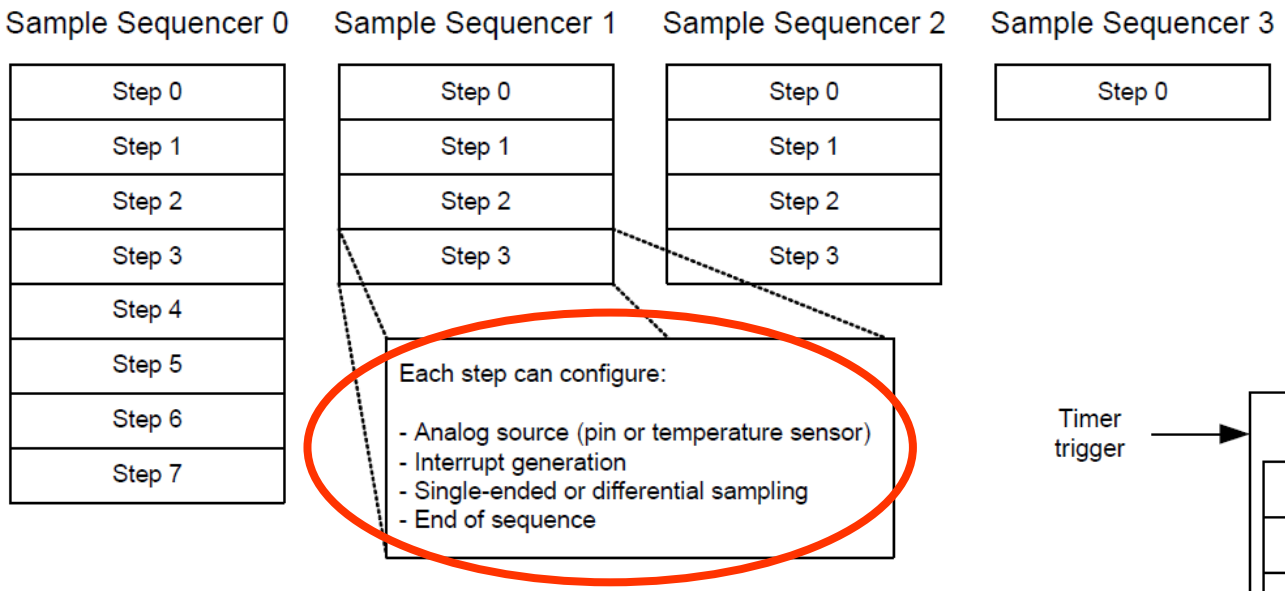
12-bit ADC Module (1)

- 2 ADC modules
  - 4 sample sequencers in each module
  - Each sequencer with configurable trigger
  - Each sequencer has 1, 4 or 8 steps
- Tiva C-series Launchpad ADC supports 8 input channels and an internal temperature sensor

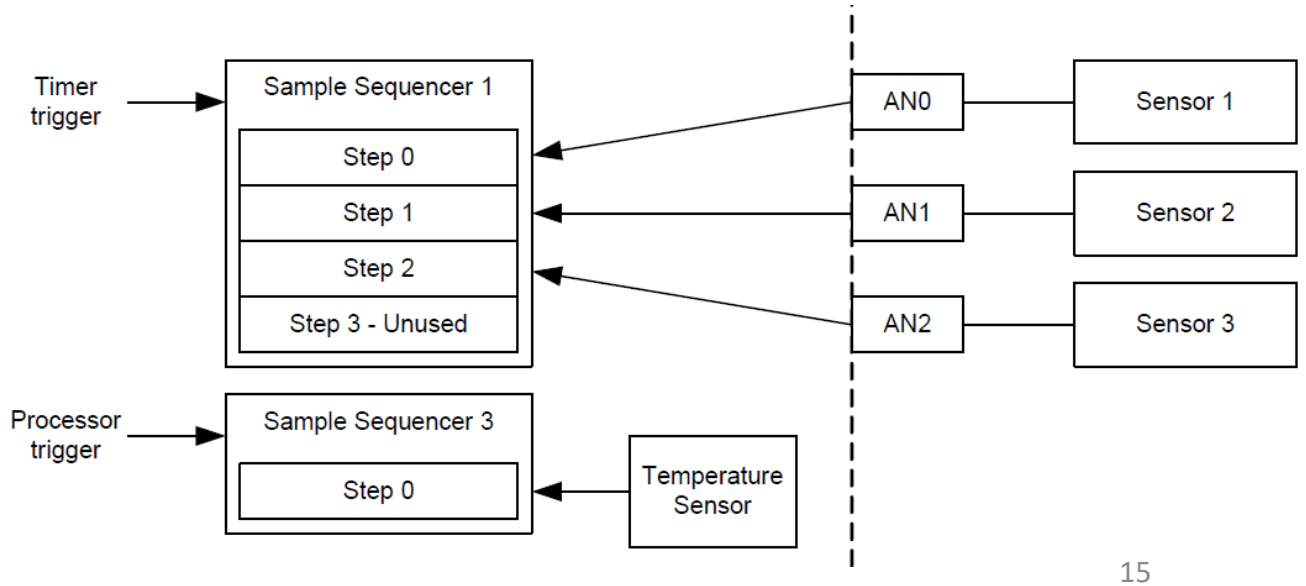


12-bit ADC Module (2)

Sample sequencer structure



Example system configuration



From Pages 4 & 5, Using the Stellaris Microcontroller Analog-to-Digital Converter (ADC).pdf

Example Code for ADC Initialization

```
void initADC (void)
{
   // Enable ADC0 module for configuration and use
   SysCtlPeripheralEnable(SYSCTL_PERIPH_ADC0);

   // Enable sample sequencer 3 with a processor signal trigger (ADC_TRIGGER_PROCESSOR)
   ADCSequenceConfigure(ADC0_BASE, 3, ADC_TRIGGER_PROCESSOR, 0);

   // Configure step 0 on sequencer 3
   // Sample channel 0 (ADC_CTL_CH0) is single-ended mode (default)
   // Configure the interrupt flag (ADC_CTL_IE) to be set when the sample is done
   // Tell the ADC logic that this is the last conversion on sequence 3 (ADC_CTL_END)
   ADCSequenceStepConfigure(ADC0_BASE, 3, 0, ADC_CTL_CH0 | ADC_CTL_IE |ADC_CTL_END);

   // Enable Sequencer 3 in ADC0
   ADCSequenceEnable(ADC0_BASE, 3);
                                                          .
                                                          .
}
```

Homework

1. Understand the example code ADCdemo1.c with help of the following documents
   - Using the Stellaris Microcontroller Analog-to-Digital Converter (ADC).pdf
   - TivaWare Peripheral Driver Library Users Manual.pdf

2. What type of ADC is used on the Tiva C-Series Launchpad?

3. What common instrument uses a dual-slope integrating ADC?

4. How many analog channels are supported by the Tiva C-Series Launchpad ADC?

5. What is the range of quantisation error in volts for an 8-bit quantiser with an input range 0 – 3 V? Repeat for 10-bit and 12-bit ADCs.

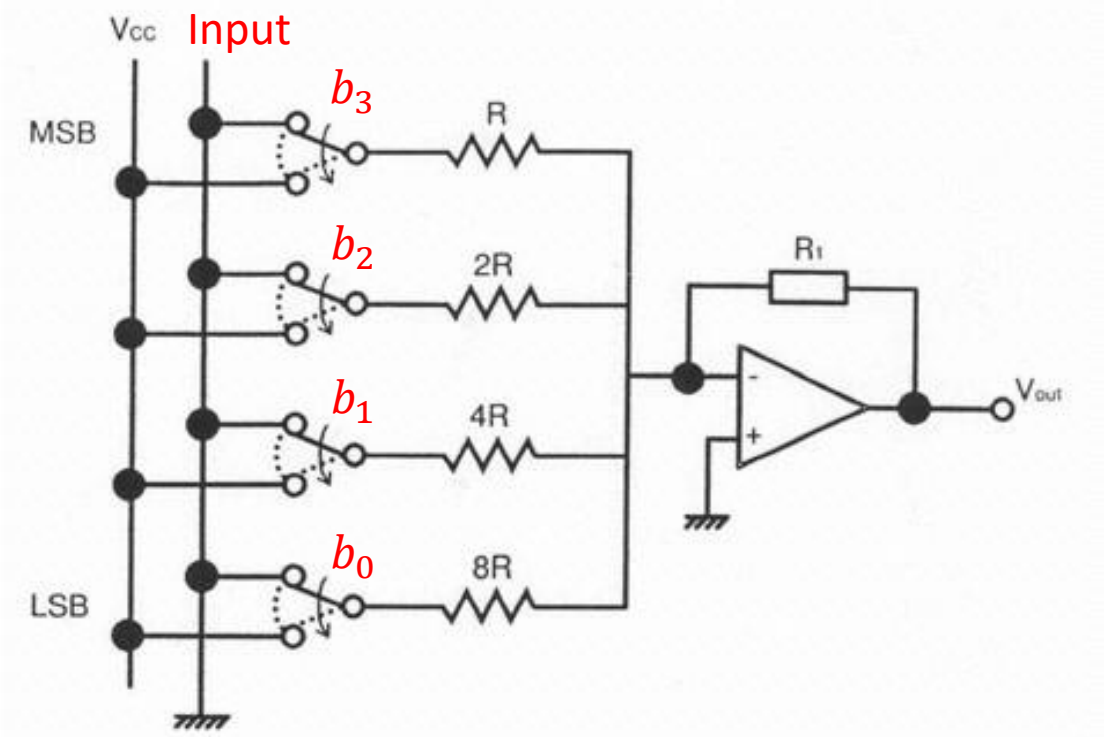
6. If the maximum signal frequency is 500 Hz and 12-bit quantization is being performed with signal averaging over 15 consecutive samples, what is the minimum sampling rate?

Digital to Analogue Conversion (DAC)

- Digital input controls switches
- Operational amplifier sums up the currents and converts to voltage for output

$$V_{out} = -V_{cc} \sum_{i=0}^{3} \frac{R_1}{2^{3-i}R} b_i$$

- DAC accuracy relies on resistances



4-bit DAC