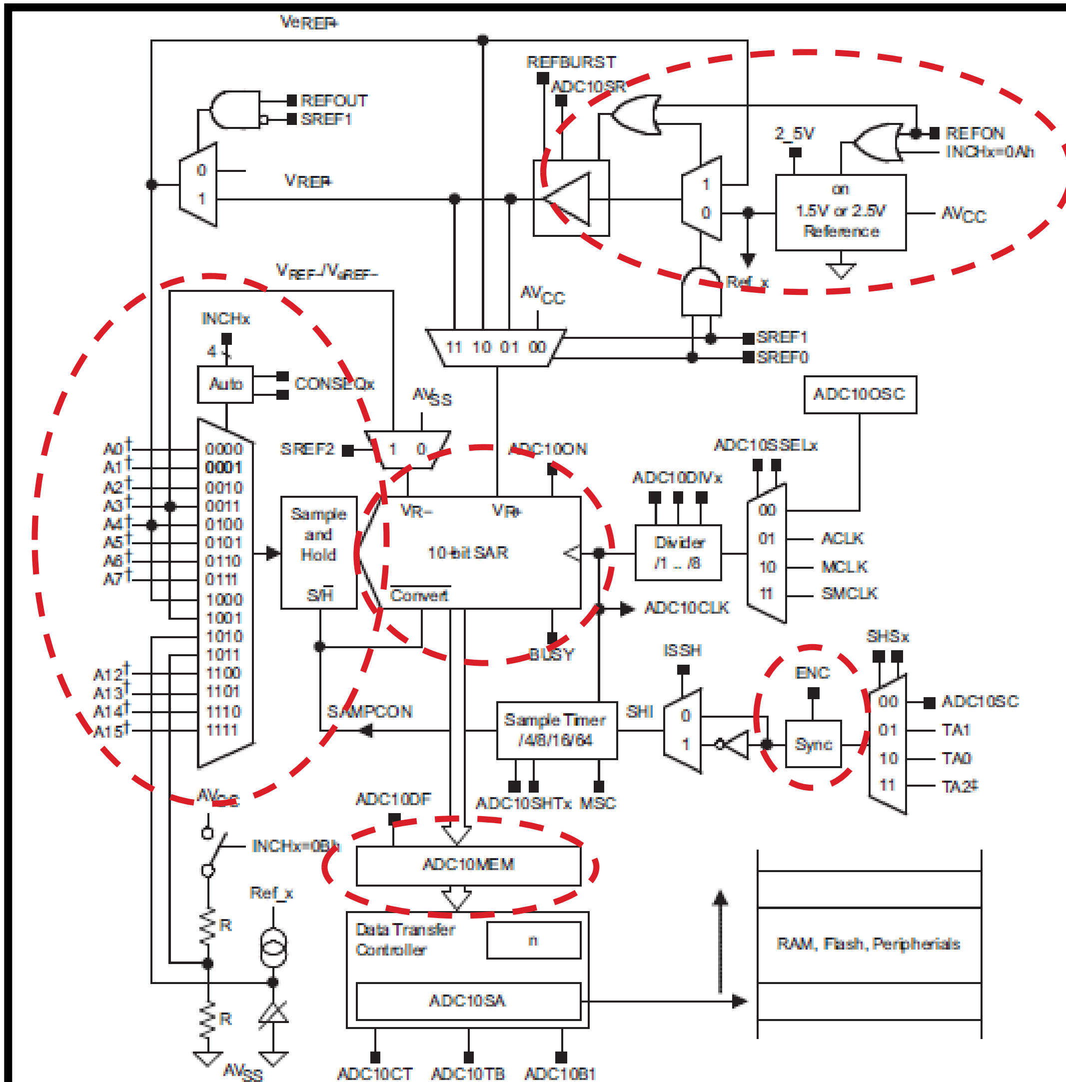
BMEN 428/ECEN 489
CSCE 489/BMEN 689

Analog-to-Digital Conversion

Please **SIGN IN** at the front!

Turn in all **POST LABS** at the front!

ADC(10) Module



ADC Registers

Table 22-3. ADC10 Registers

Register	Short Form	Register Type	Address	Initial State
ADC10 input enable register 0	ADC10AE0	Read/write	04Ah	Reset with POR
ADC10 input enable register 1	ADC10AE1	Read/write	04Bh	Reset with POR
ADC10 control register 0	ADC10CTL0	Read/write	01B0h	Reset with POR
ADC10 control register 1	ADC10CTL1	Read/write	01B2h	Reset with POR
ADC10 memory	ADC10MEM	Read	01B4h	Unchanged
ADC10 data transfer control register 0	ADC10DTC0	Read/write	048h	Reset with POR
ADC10 data transfer control register 1	ADC10DTC1	Read/write	049h	Reset with POR
ADC10 data transfer start address	ADC10SA	Read/write	01BCh	0200h with POR

Use the MSP430x2xx Family User Guide as a reference!

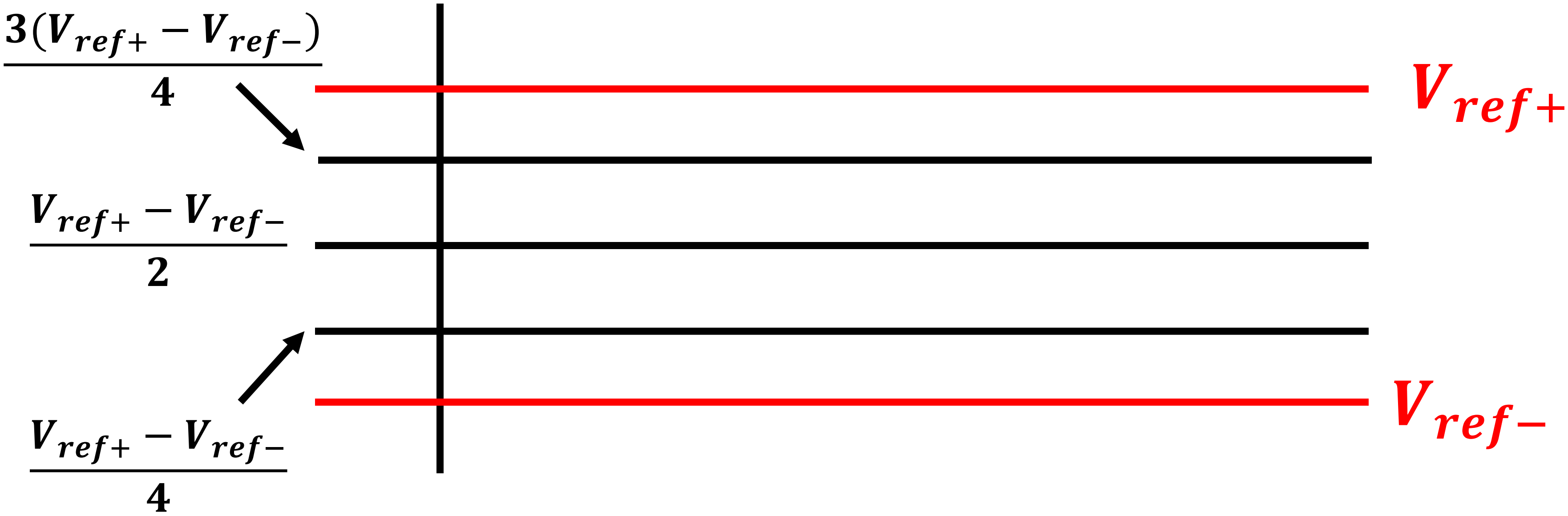
Some of these registers may or may not necessarily be used!

How a typical ADC Works…

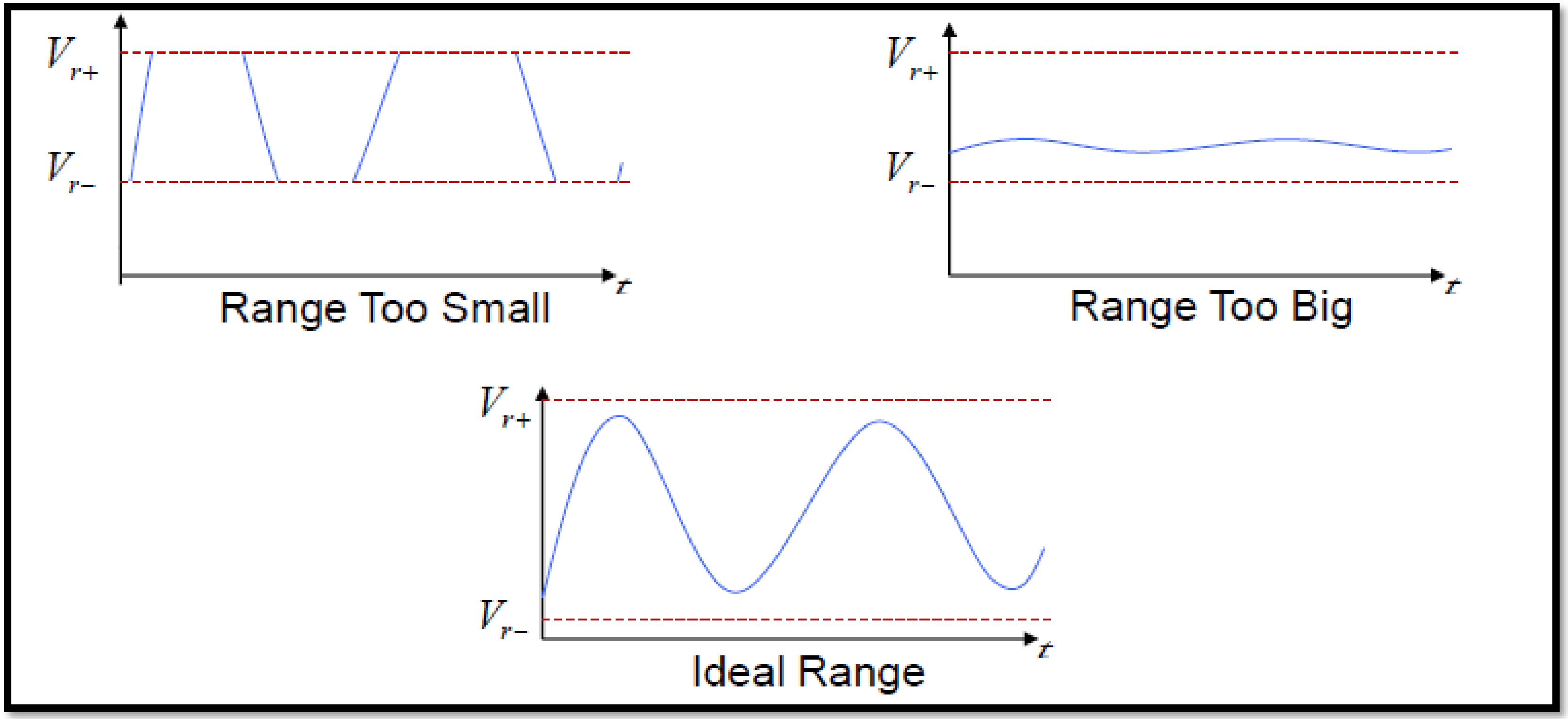
Example: a 2-bit ADC

a 2-bit ADC = 2^2 = 4 divisions between reference voltages

$$resolution = \frac{V_{ref+} - V_{ref-}}{4}$$



Selecting the Reference Voltages

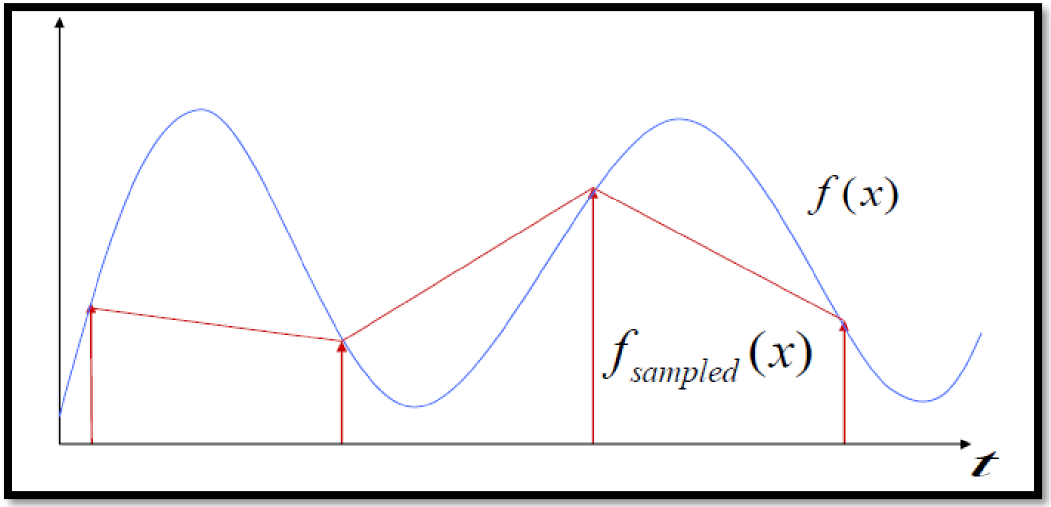


By adjusting the reference voltages (V_{r+} and V_{r-}), you can adjust the resolution of the ADC.

Selecting the Sampling Rate

- **What sample rate do we need?**

- **Too little: we can't reconstruct the signal we care about.**
- **Too much: waste computation, energy, resources**



Use the correct register to select the sampling rate!

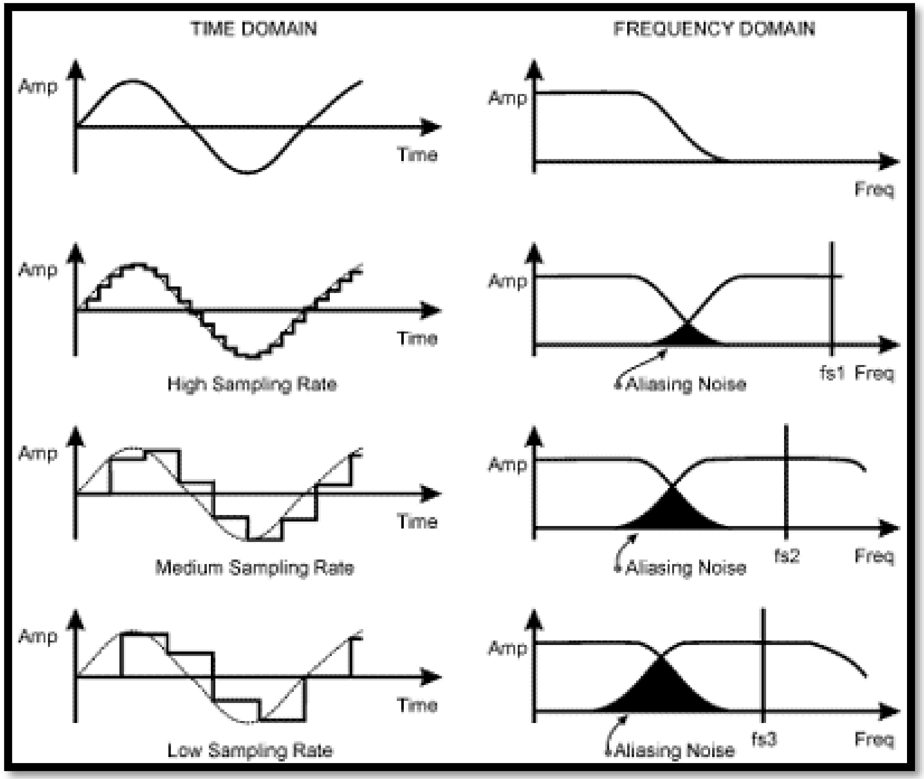
***Hint*: 2 options only.**

Shannon-Nyquist Theorem

- ***If a continuous-time signal contains no frequencies higher than , it can be completely determined by discrete samples taken at a rate:***

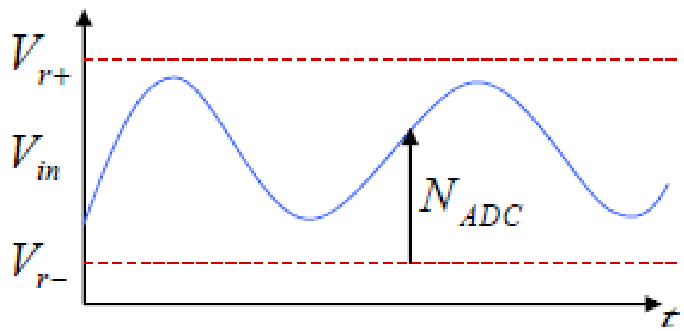
$$f_{\text{samples}} > 2 f_{\max}$$

- ***Beware of the bandwidth of the input signal. If the bandwidth is too large, the sampling process may require an anti-aliasing filter.***



Converting between ADC Counts and Voltage

- Converting: ADC counts ⇔ Voltage



$$N_{ADC} = 4095 \times \frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}}$$

$$V_{in} = V_{R-} + N_{ADC} \times \frac{V_{R+} - V_{R-}}{4095}$$

Use the lecture notes as a reference!

Shortcuts for ADC Registers

22.3.1 ADC10CTL0, ADC10 Control Register 0

15	14	13	12	11	10	9	8
SREFx			ADC10SHTx		ADC10SR	REFOUT	REFBURST
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC10ON	ADC10IE	ADC10IFG	ENC	ADC10SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Can be modified only when ENC = 0

SREFx Bits 15-13 Select reference.

- 000 $V_{R+} = V_{CC}$ and $V_{R-} = V_{SS}$
- 001 $V_{R+} = V_{REF+}$ and $V_{R-} = V_{SS}$
- 010 $V_{R+} = V_{eREF+}$ and $V_{R-} = V_{SS}$. Devices with V_{eREF+} only.
- 011 V_{R+} = Buffered V_{eREF+} and $V_{R-} = V_{SS}$. Devices with V_{eREF+} pin only.
- 100 $V_{R+} = V_{CC}$ and $V_{R-} = V_{REF-}/ V_{eREF-}$. Devices with V_{eREF-} pin only.
- 101 $V_{R+} = V_{REF+}$ and $V_{R-} = V_{REF-}/ V_{eREF-}$. Devices with $V_{eREF+/-}$ pins only.
- 110 $V_{R+} = V_{eREF+}$ and $V_{R-} = V_{REF-}/ V_{eREF-}$. Devices with $V_{eREF+/-}$ pins only.
- 111 V_{R+} = Buffered V_{eREF+} and $V_{R-} = V_{REF-}/ V_{eREF-}$. Devices with $V_{eREF+/-}$ pins only.

ADC10CTL0 is a 16-bit register.

Shortcuts for ADC Registers

22.3.1 ADC10CTL0, ADC10 Control Register 0

15	14	13	12	11	10	9	8
	SREFx		ADC10SHTx		ADC10SR	REFOUT	REFBURST
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC10ON	ADC10IE	ADC10IFG	ENC	ADC10SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Can be modified only when ENC = 0

SREFx Bits 15-13 Select reference.

- 000 $V_{R+} = V_{CC}$ and $V_{R-} = V_{SS}$
- 001 $V_{R+} = V_{REF+}$ and $V_{R-} = V_{SS}$
- 010 $V_{R+} = V_{eREF+}$ and $V_{R-} = V_{SS}$. Devices with V_{eREF+} only.
- 011 V_{R+} = Buffered V_{eREF+} and $V_{R-} = V_{SS}$. Devices with V_{eREF+} pin only.
- 100 $V_{R+} = V_{CC}$ and $V_{R-} = V_{REF-}/ V_{eREF-}$. Devices with V_{eREF-} pin only.
- 101 $V_{R+} = V_{REF+}$ and $V_{R-} = V_{REF-}/ V_{eREF-}$. Devices with $V_{eREF+/-}$ pins only.
- 110 $V_{R+} = V_{eREF+}$ and $V_{R-} = V_{REF-}/ V_{eREF-}$. Devices with $V_{eREF+/-}$ pins only.
- 111 V_{R+} = Buffered V_{eREF+} and $V_{R-} = V_{REF-}/ V_{eREF-}$. Devices with $V_{eREF+/-}$ pins only.

ADC10CTL0 |= BIT15 + BIT14 + BIT13;

ADC10CTL0 |= BIT15 | BIT14 | BIT13;

ADC10CTL0 |= SREF_7;

A word about shortcuts…

```
ADC10CTL0 |= BIT15 + BIT14 + BIT13;
ADC10CTL0 |= BIT15 | BIT14 | BIT13;
ADC10CTL0 |= SREF_7;
```

Setting ADC10CTL0 =

"<u>111</u>XXXXXXXXXXXXX"

bitwise operations!

You can also manipulate the register in ***<u>it's entirety</u>***:

```
ADC10CTL0 = BIT15 + BIT14 + BIT13;
ADC10CTL0 = BIT15 | BIT14 | BIT13;
ADC10CTL0 = SREF_7;
```

Setting ADC10CTL0 =

"<u>111</u>0000000000000"

ADC Set-Up

Two ways to approach the problem:

1) Continuous sampling?
   - Need to define registers for continuous sampling.

2) LPW and interrupt?
   - Default mode, read and sample JUST one input data point.
   - Solution?

Refer to family user guide as a reference for the different sampling modes!

ADC Set-Up

A SOFTWARE view (LPM & Interrupt):

1. Set some control registers :
   - Specify where the input is coming from (which pin).
   - Specify the range (min and max).
   - Specify reference voltages (if needed).

2. Enable interrupt and start conversion.
   - Set µC into LPM.

3. When interrupt occurs, read input from data register.
4. Output?
5. Exit LPM.

Expected Result