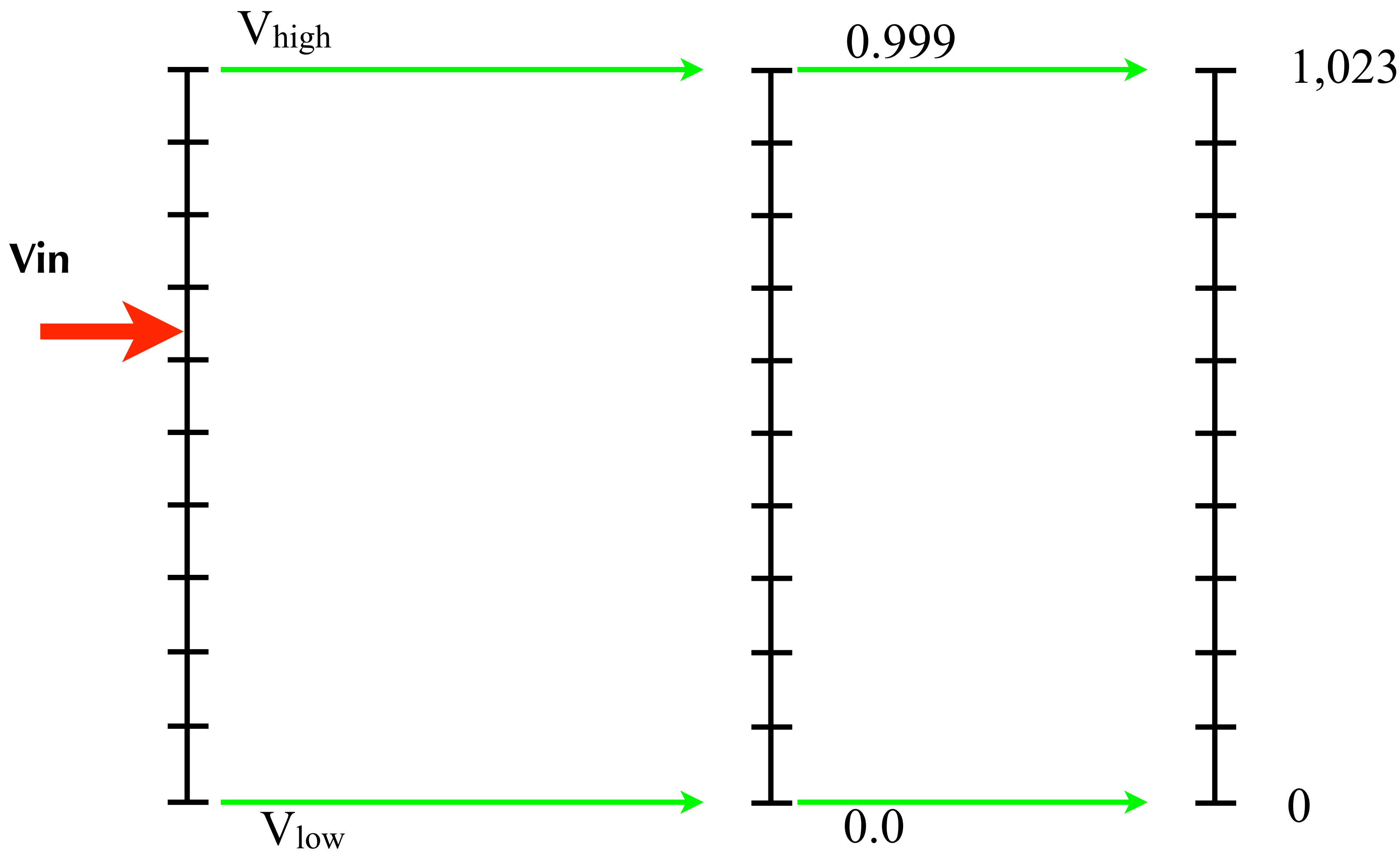
Analog to Digital Conversion

Gary J. Minden
October 1, 2013

Information and Telecommunication Technology Center

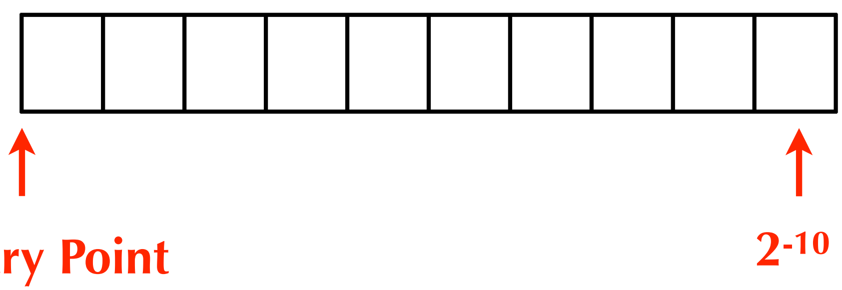
THE UNIVERSITY OF KANSAS

Mapping Input Voltage to Digital Value

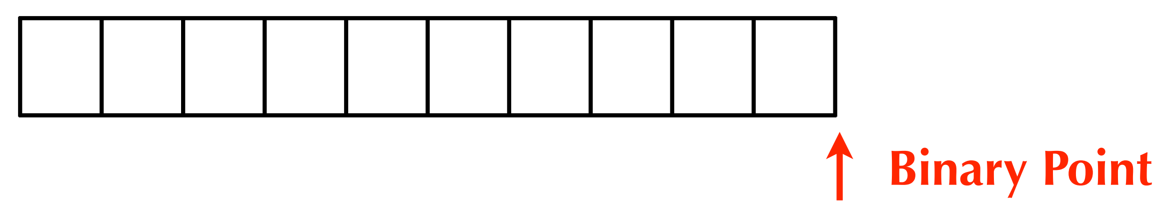


Analog to Digital Conversion

- Analog -- A voltage between V_{low} and V_{high}
- Digital -- A fraction between 0.0 and 1.0 with K bits
  - In the case of the LM3S1968, K = 10 and there are 1,024 steps.
  - Fraction: Values between 0.0 and 0.999023



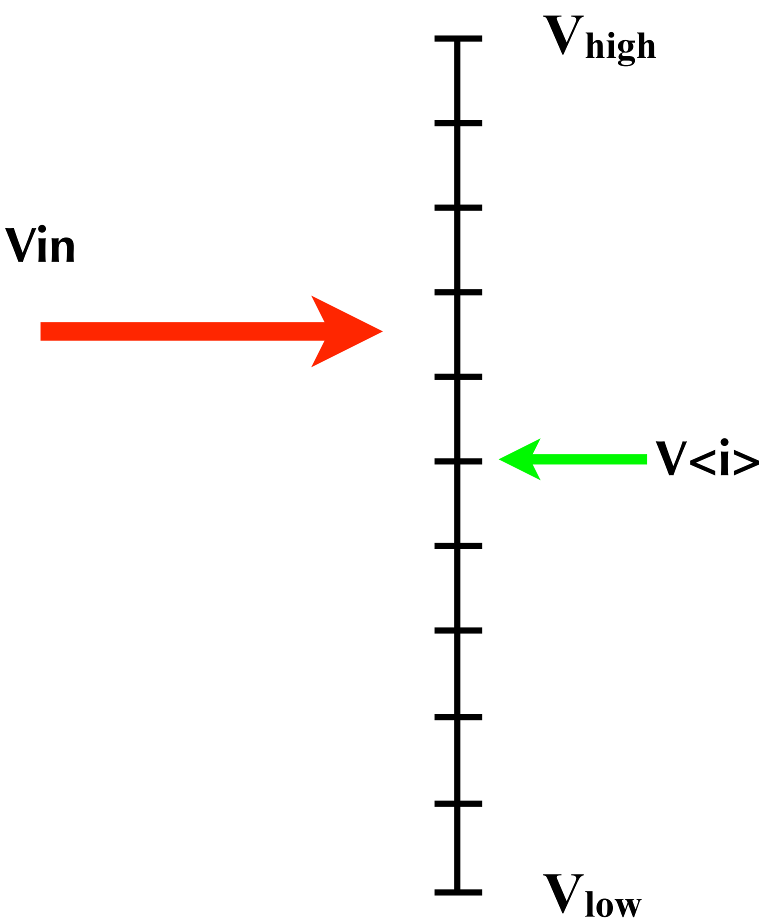
  - Integer: Values between 0 and 1,023 with scale factor of 2^{-10}



Information and Telecommunication Technology Center

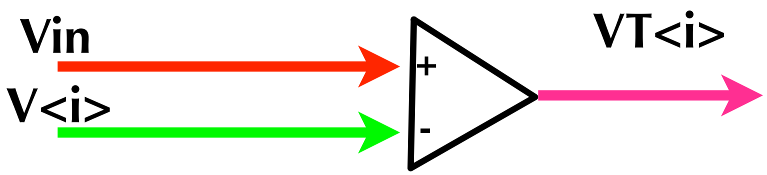
THE UNIVERSITY OF KANSAS

Challenge #1 -- Compare Voltage



- Vin is input Voltage
- Range is $\mathbf{V_{low}}$ to $\mathbf{V_{high}}$
- **V<i>** is "test" Voltage
- How to compare Vin to **V<i>**?
- That is, is Vin < **V<i>** or not?

Comparator

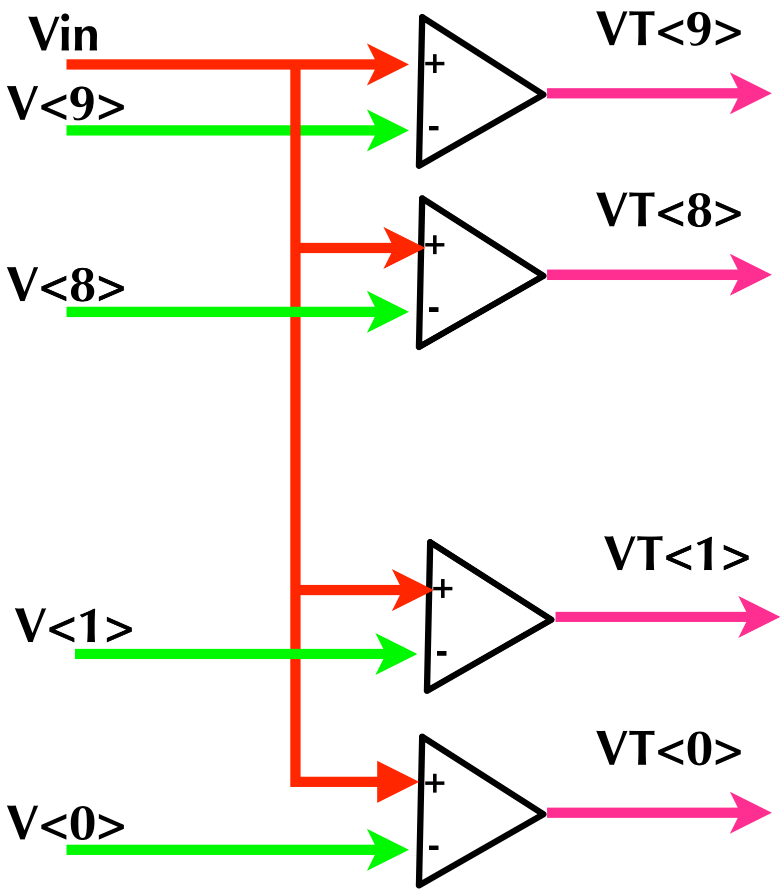


- If Vin > **V<i>**, then VT<i> is True (1)
- A logic signal

Information and Telecommunication Technology Center

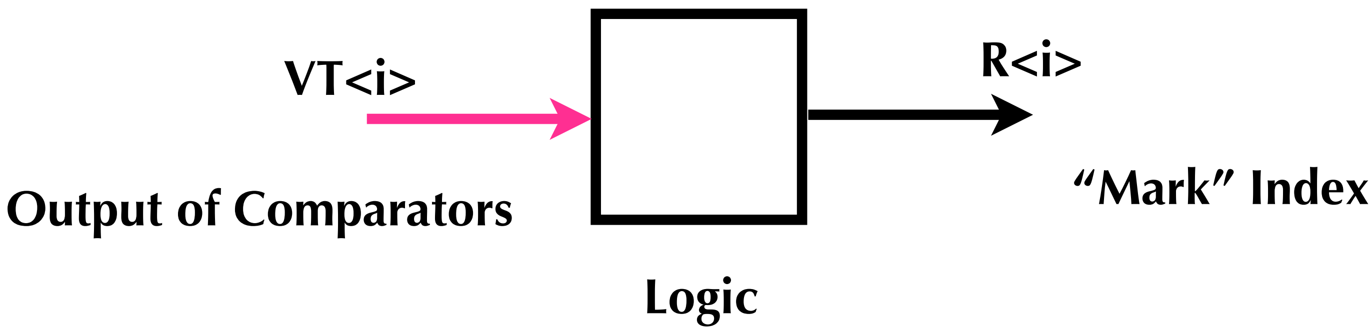
THE UNIVERSITY OF KANSAS

Array of Comparators



- Set up array of comparators
- Each comparator tests Vin
- Each has a different **V<i>**
- Each comparator generates a logic value, 0 or 1

Results

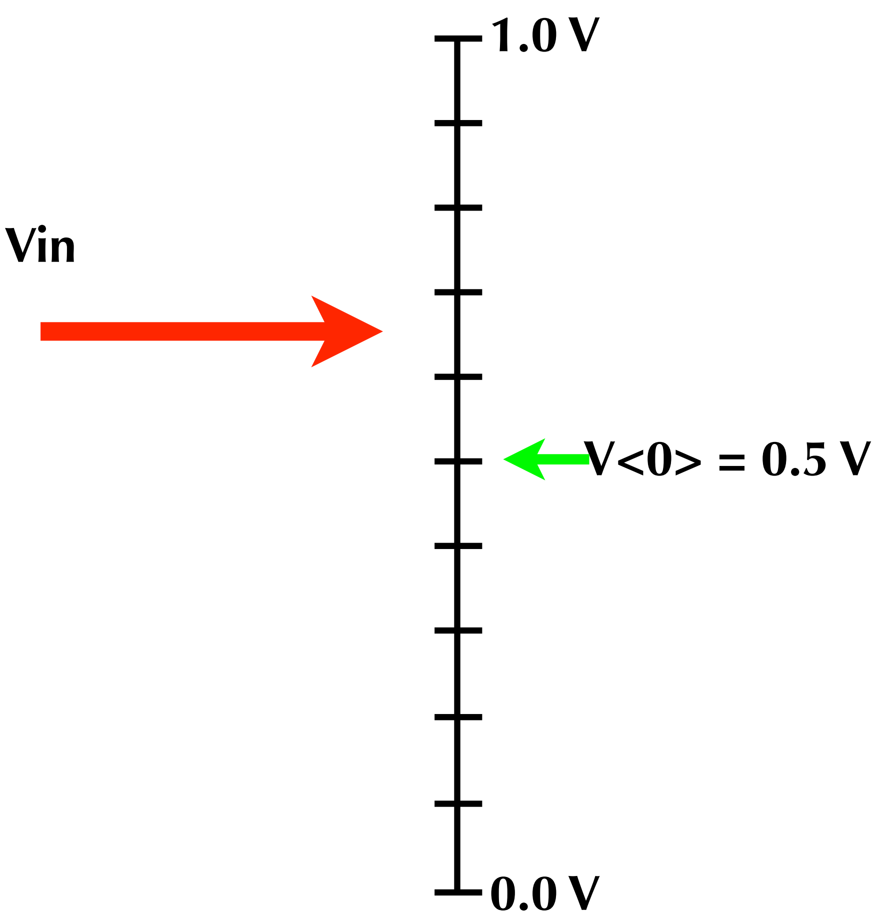


V<9>	V<8>	V<7>	V<6>	V<5>	V<4>	V<3>	V<2>	V<1>	V<0>	R<3>	R<2>	R<1>	R<0>
0	0	0	0	1	1	1	1	1	1	0	1	0	1
0	0	0	0	0	0	0	1	1	1	0	0	1	0
0	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Problems and Approach

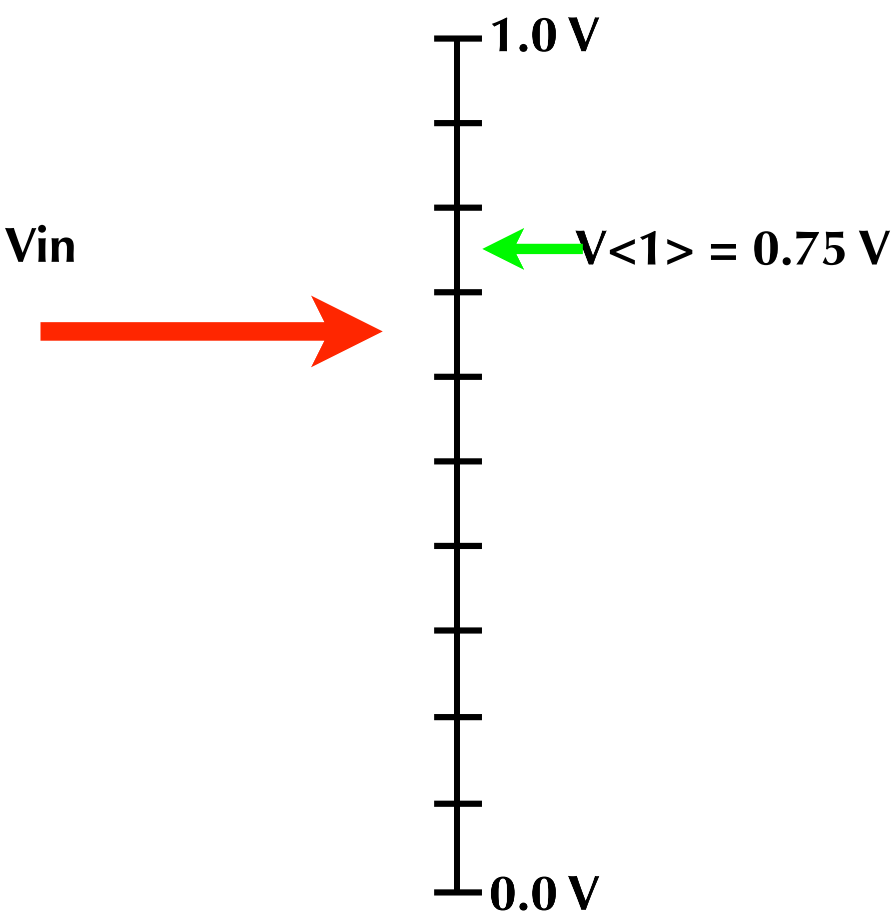
- This does not scale for a large number of “steps”
  - E.g. 1,024 or 4,096
- Successively generate V<i>, a test Voltage, over several trials
  - What sequence of V<i> to use
  - How do you generate V<i>?

Successive Approximation Approach #1



- Vin is input Voltage
- Range is 0.0 - 1.0 V
- Assume 1,024 steps
- V<0> is first test Voltage
- Start in the middle
- If Vin < V<0>, decrease test voltage
- If Vin >= V<0>, increase test voltage
- **Vin >** V<0>**, increase**

Successive Approximation Approach #2

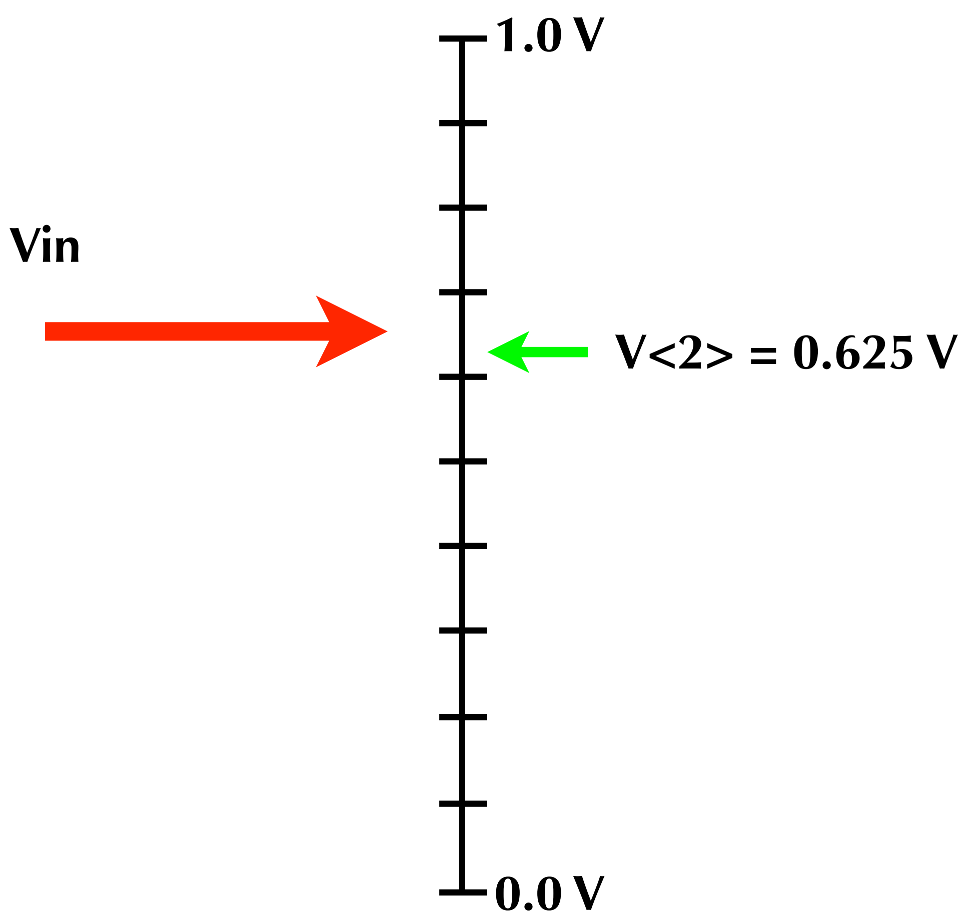


- Vin is input Voltage
- Range is 0.0 - 1.0 V
- Assume 1,024 steps
- V<1> is test voltage
- If Vin < V<1>, decrease test voltage
- If Vin >= V<1>, increase test voltage
- **Vin < V<1>, decrease test voltage**

Information and Telecommunication Technology Center

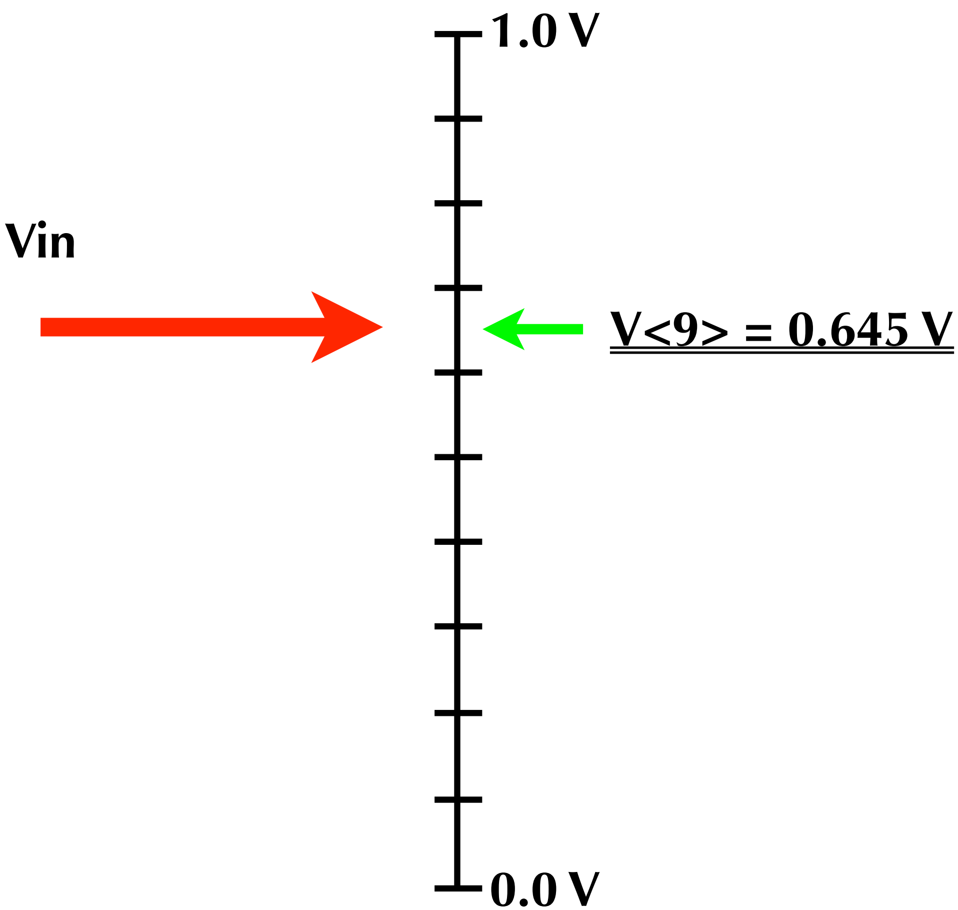
THE UNIVERSITY OF KANSAS

Successive Approximation Approach #3



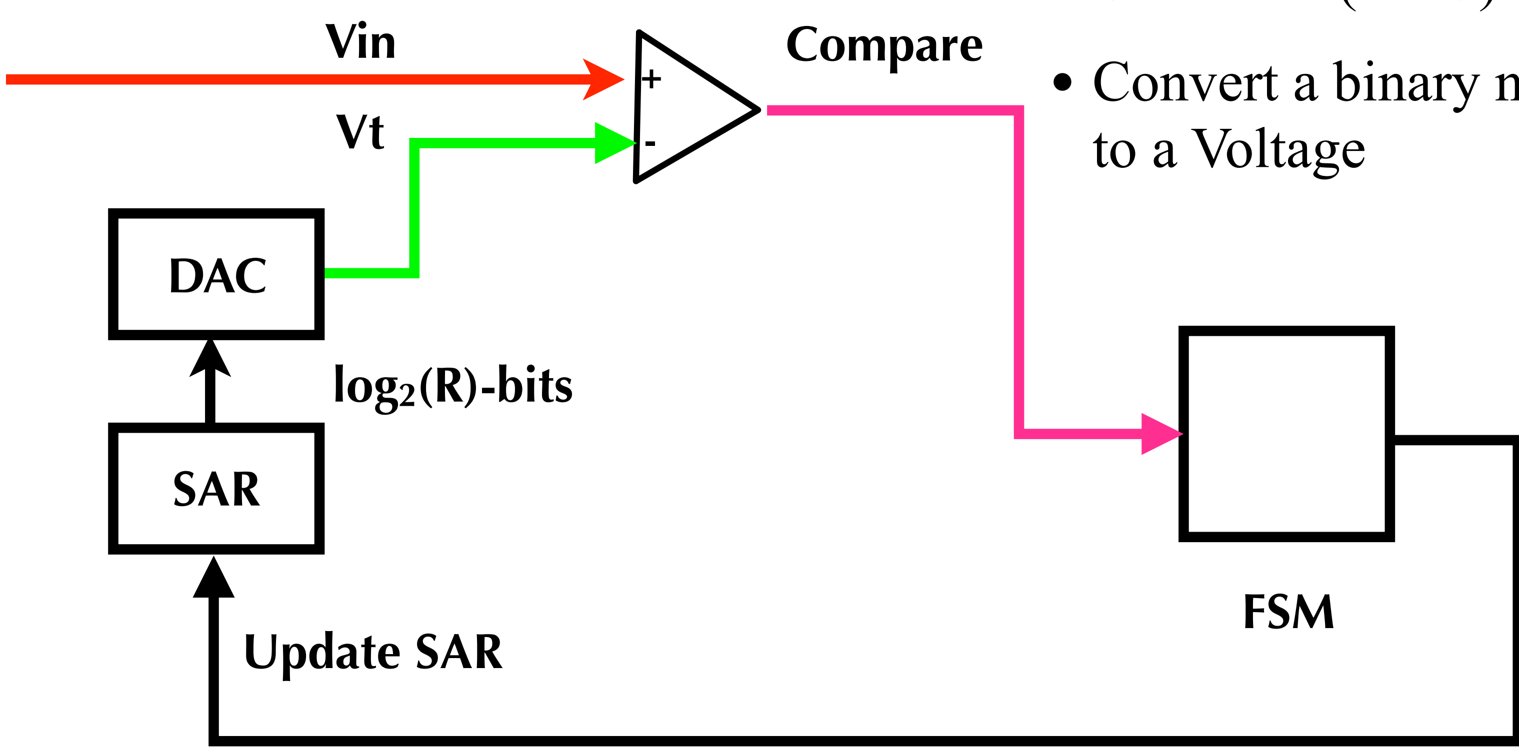
- Vin is input Voltage
- Range is 0.0 - 1.0 V
- Assume 1,024 steps
- V<2> is test voltage
- Start in the middle
- If Vin < V<2>, decrease V<i>
- If Vin >= V<2>, increase V<i>
- **Vin >= V<2>, increase**

Successive Approximation Approach #10



- Vin is input Voltage
- Range is 0.0 - 1.0 V
- Assume 1,024 steps
- V<9> is test Voltage
- If Vin < V<9>, decrease test voltage
- If Vin >= V<9>, increase test voltage
- **Done**

Successive Approach with DAC



- Generate V<i> with a Digital to Analog Converter (DAC)
- Convert a binary number to a Voltage

Information and Telecommunication Technology Center

THE UNIVERSITY OF KANSAS

Example with Vin = 0.644 V

i	S<9>	S<8>	S<7>	S<6>	S<5>	S<4>	S<3>	S<2>	S<1>	S<0>	V<i>	Comp
0	1	0	0	0	0	0	0	0	0	0	0.5	1
1	1	1	0	0	0	0	0	0	0	0	0.75	0
2	1	0	1	0	0	0	0	0	0	0	0.625	1
3	1	0	1	1	0	0	0	0	0	0	0.688	0
4	1	0	1	0	1	0	0	0	0	0	0.656	0
5	1	0	1	0	0	1	0	0	0	0	0.641	1
6	1	0	1	0	0	1	1	0	0	0	0.648	0
7	1	0	1	0	0	1	0	1	0	0	0.645	0
8	1	0	1	0	0	1	0	0	1	0	0.643	1
9	1	0	1	0	0	1	0	0	1	1	0.644	1
D	1	0	1	0	0	1	0	0	1	1	0.644	

Compare == 1 if Vin >= V<i>

Example with Vin = 0.644 V

i	S<9>	S<8>	S<7>	S<6>	S<5>	S<4>	S<3>	S<2>	S<1>	S<0>	V<i>	Comp
0	1	0	0	0	0	0	0	0	0	0	0.5	1
1	1	1	0	0	0	0	0	0	0	0	0.75	0
2	1	0	1	0	0	0	0	0	0	0	0.625	1
3	1	0	1	1	0	0	0	0	0	0	0.688	0
4	1	0	1	0	1	0	0	0	0	0	0.656	0
5	1	0	1	0	0	1	0	0	0	0	0.641	1
6	1	0	1	0	0	1	1	0	0	0	0.648	0
7	1	0	1	0	0	1	0	1	0	0	0.645	0
8	1	0	1	0	0	1	0	0	1	0	0.643	1
9	1	0	1	0	0	1	0	0	1	1	0.644	1
D	1	0	1	0	0	1	0	0	1	1	0.644	

Compare == 1 if Vin >= V<i>

Example with Vin = 0.644 V

i	S<9>	S<8>	S<7>	S<6>	S<5>	S<4>	S<3>	S<2>	S<1>	S<0>	V<i>	Comp
0	1	0	0	0	0	0	0	0	0	0	0.5	1
1	1	1	0	0	0	0	0	0	0	0	0.75	0
2	1	0	1	0	0	0	0	0	0	0	0.625	1
3	1	0	1	1	0	0	0	0	0	0	0.688	0
4	1	0	1	0	1	0	0	0	0	0	0.656	0
5	1	0	1	0	0	1	0	0	0	0	0.641	1
6	1	0	1	0	0	1	1	0	0	0	0.648	0
7	1	0	1	0	0	1	0	1	0	0	0.645	0
8	1	0	1	0	0	1	0	0	1	0	0.643	1
9	1	0	1	0	0	1	0	0	1	1	0.644	1
D	1	0	1	0	0	1	0	0	1	1	0.644	

Compare == 1 if Vin >= V<i>

Example with Vin = 0.644 V

i	S<9>	S<8>	S<7>	S<6>	S<5>	S<4>	S<3>	S<2>	S<1>	S<0>	V<i>	Comp
0	1	0	0	0	0	0	0	0	0	0	0.5	1
1	1	1	0	0	0	0	0	0	0	0	0.75	0
2	1	0	1	0	0	0	0	0	0	0	0.625	1
3	1	0	1	1	0	0	0	0	0	0	0.688	0
4	1	0	1	0	1	0	0	0	0	0	0.656	0
5	1	0	1	0	0	1	0	0	0	0	0.641	1
6	1	0	1	0	0	1	1	0	0	0	0.648	0
7	1	0	1	0	0	1	0	1	0	0	0.645	0
8	1	0	1	0	0	1	0	0	1	0	0.643	1
9	1	0	1	0	0	1	0	0	1	1	0.644	1
D	1	0	1	0	0	1	0	0	1	1	0.644	

Compare == 1 if Vin >= V<i>

Example with Vin = 0.644 V

i	S<9>	S<8>	S<7>	S<6>	S<5>	S<4>	S<3>	S<2>	S<1>	S<0>	V<i>	Comp
0	1	0	0	0	0	0	0	0	0	0	0.5	1
1	1	1	0	0	0	0	0	0	0	0	0.75	0
2	1	0	1	0	0	0	0	0	0	0	0.625	1
3	1	0	1	1	0	0	0	0	0	0	0.688	0
4	1	0	1	0	1	0	0	0	0	0	0.656	0
5	1	0	1	0	0	1	0	0	0	0	0.641	1
6	1	0	1	0	0	1	1	0	0	0	0.648	0
7	1	0	1	0	0	1	0	1	0	0	0.645	0
8	1	0	1	0	0	1	0	0	1	0	0.643	1
9	1	0	1	0	0	1	0	0	1	1	0.644	1
D	1	0	1	0	0	1	0	0	1	1	0.644	

Compare == 1 if Vin >= V<i>

Result Value Format

Scale Factor

2^-1	2^-2	2^-3	2^-4	2^-5	2^-6	2^-7	2^-8	2^-9	2^-10
1/2	1/4	1/8	1/16	1/32	1/64	1/128	1/256	1/512	1/1024
1	0	1	0	0	1	0	0	1	1

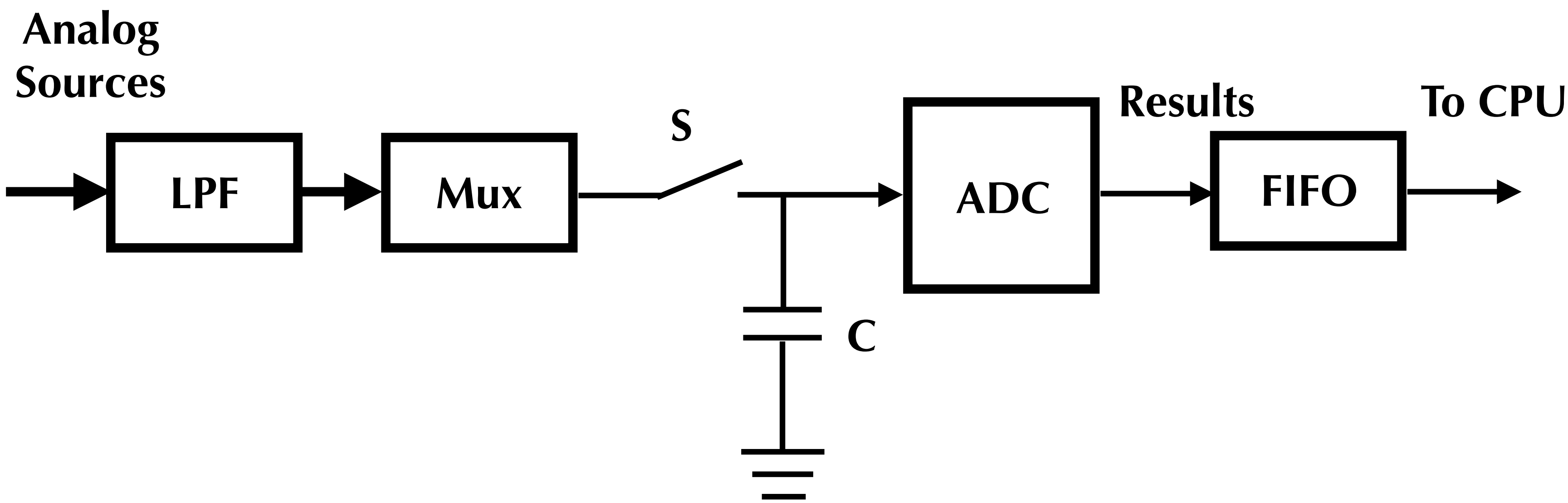
Binary Point

$$R = 1/2 + 1/8 + 1/64 + 1/512 + 1/1024 = 0.644$$

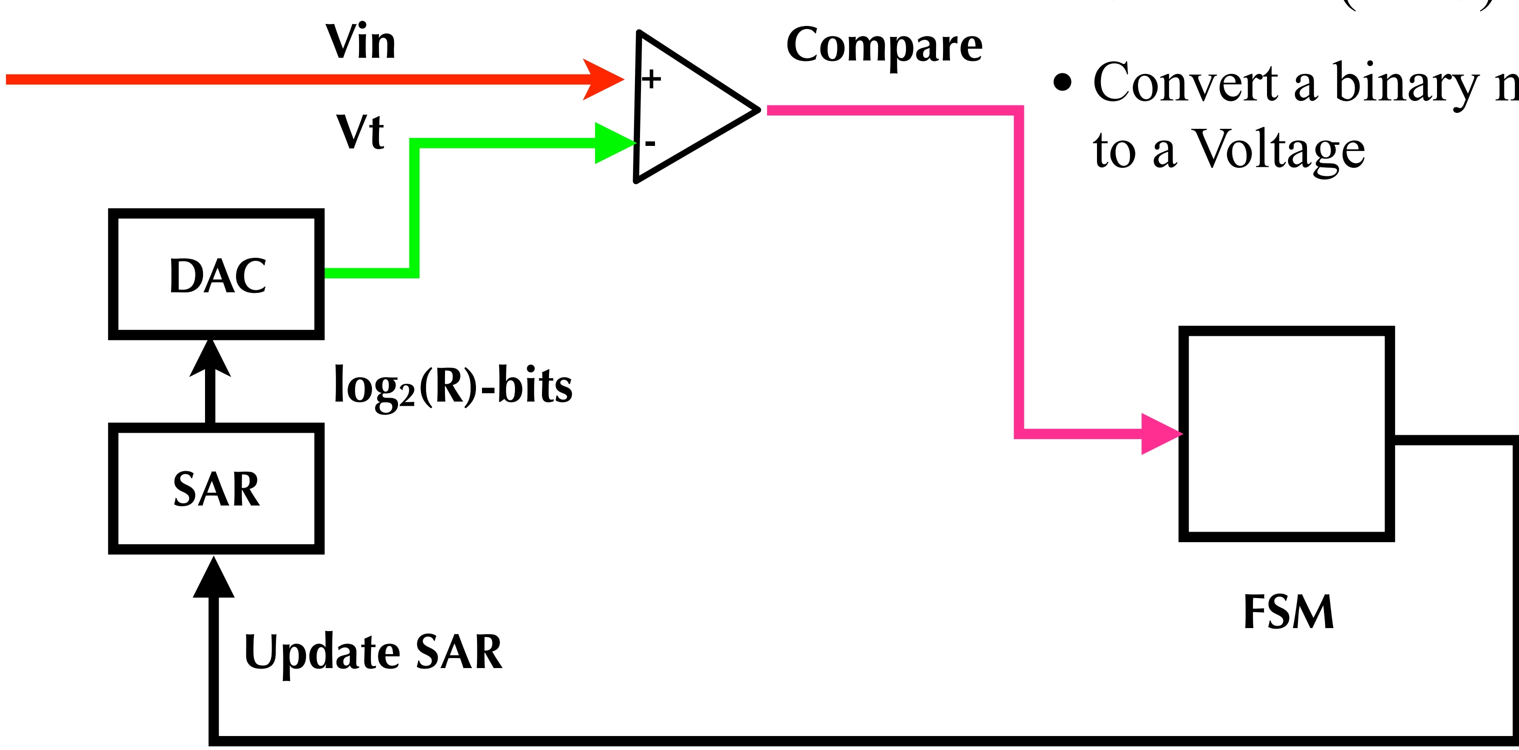
Information and Telecommunication Technology Center

THE UNIVERSITY OF KANSAS

ADC Architecture



Successive Approach with DAC

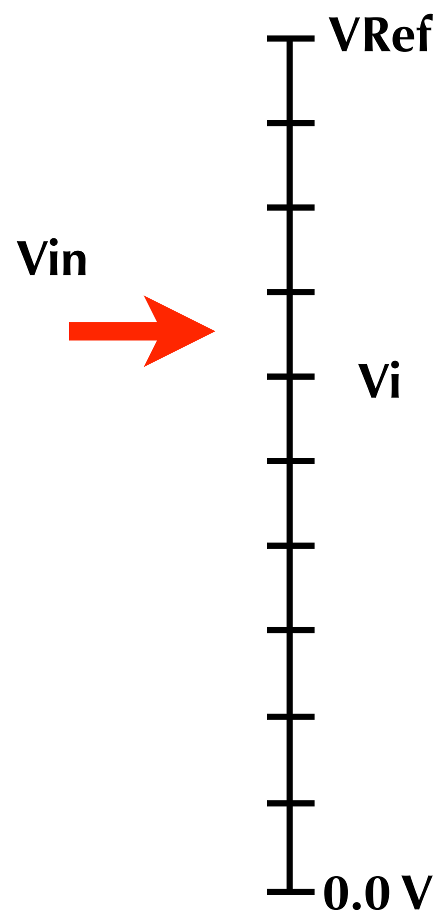


- Generate V<i> with a Digital to Analog Converter (DAC)
- Convert a binary number to a Voltage

Measured Voltage

$V_{Measured} = R * V_{Ref}$

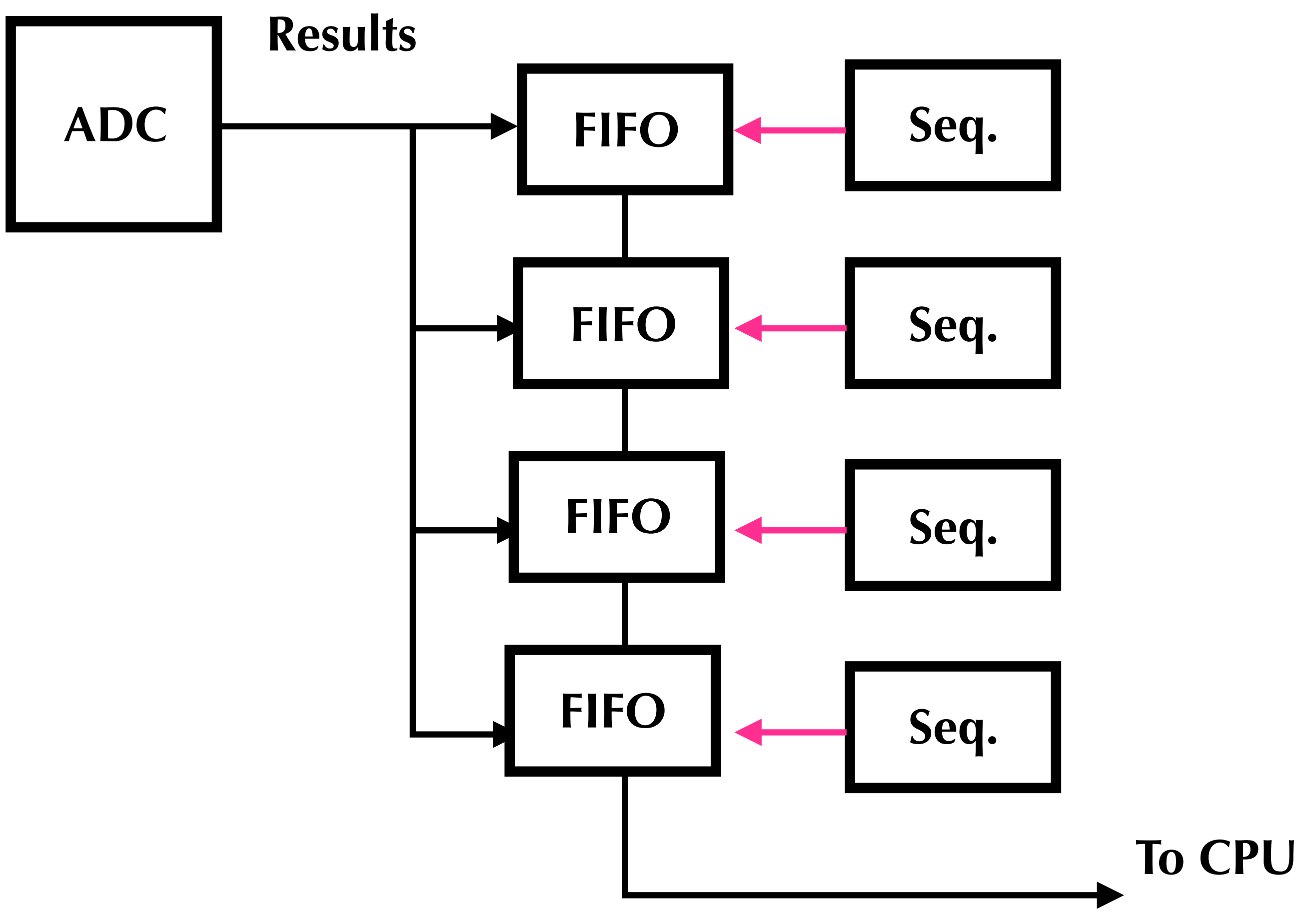
Internal $V_{Ref} = 3.0$ V



LM3S1968 ADC Features

- Eight analog input channels
- Single-ended and differential-input configurations
- On-chip internal temperature sensor
- Sample rate of one million samples/second
- Four programmable sample conversion sequences from one to eight entries long, with corresponding conversion result FIFOs
- Flexible trigger control
  - Controller (software), Timers, Analog Comparators, PWM, GPIO
- Hardware averaging of up to 64 samples for improved accuracy
- Converter uses an internal 3-V reference
- Power and ground for the analog circuitry is separate from the digital power and ground

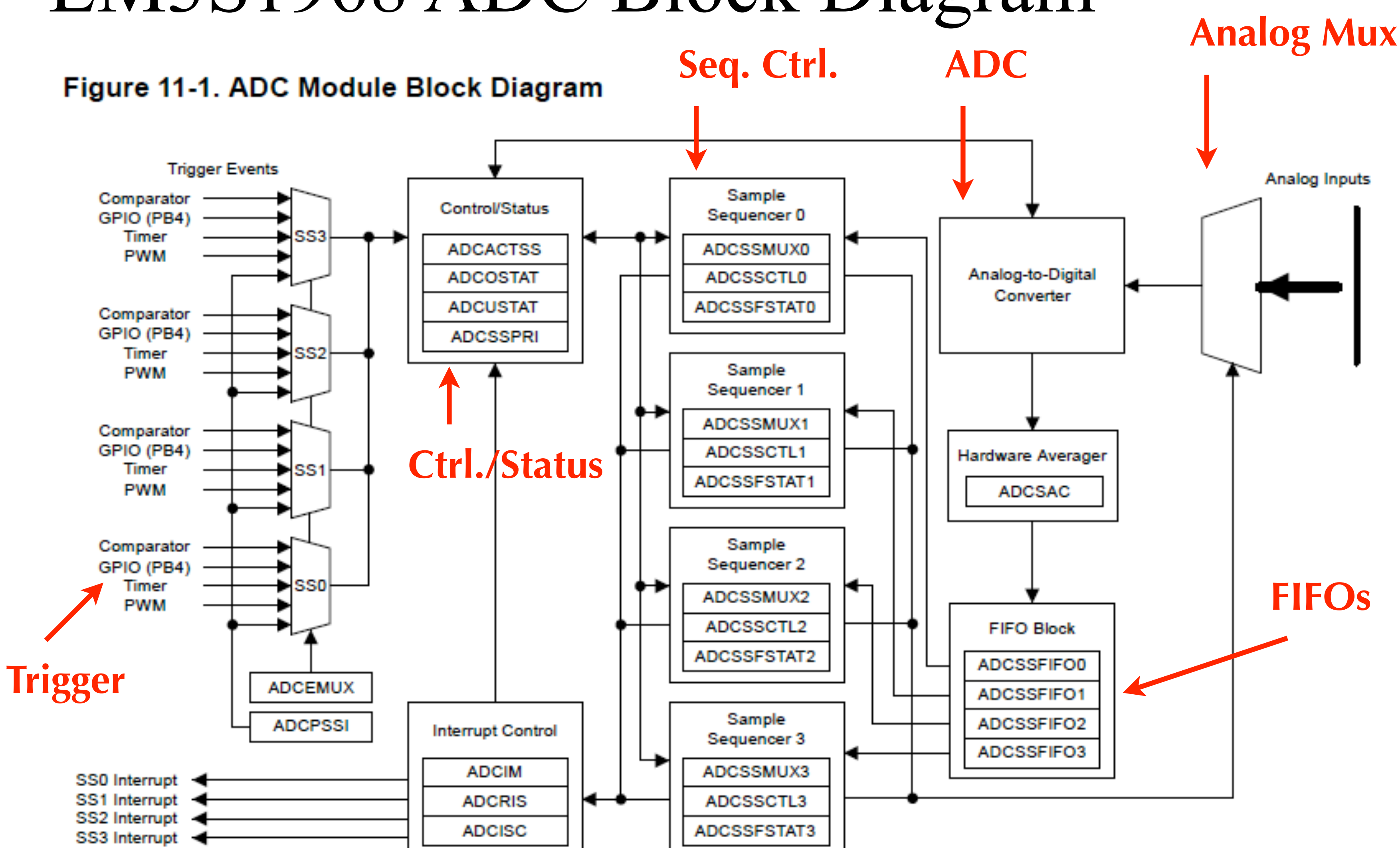
Stellaris ADC Architecture



ADC Control

- Enable
- Interrupt generation
- DMA operation
- Sequence prioritization
  - Each sequencer has a priority, 0-3, 0 is highest
- Trigger configuration
  - Processor (default), analog comparators, an external signal on GPIO PB4 , a GP Timer, a PWM generator, and continuous sampling.
- Comparator configuration
- External voltage reference
- Sample phase control

LM3S1968 ADC Block Diagram



Information and Telecommunication Technology Center

THE UNIVERSITY OF KANSAS

ADC Sequencer Characteristics

Table 11-3. Samples and FIFO Depth of Sequencers

Sequencer	Number of Samples	Depth of FIFO
SS3	1	1
SS2	4	4
SS1	4	4
SS0	8	8

Most of the ADC control logic runs at the ADC clock rate of 14-18 MHz. The internal ADC divider is configured automatically by hardware when the system `XTAL` is selected. The automatic clock divider configuration targets 16.667 MHz operation for all Stellaris devices.

Differential Inputs

Table 11-4. Differential Sampling Pairs

Differential Pair	Analog Inputs
0	0 and 1
1	2 and 3
2	4 and 5
3	6 and 7

The voltage sampled in differential mode is the difference between the odd and even channels:

ΔV (differential voltage) = V_{IN_EVEN} (even channels) – V_{IN_ODD} (odd channels), therefore:

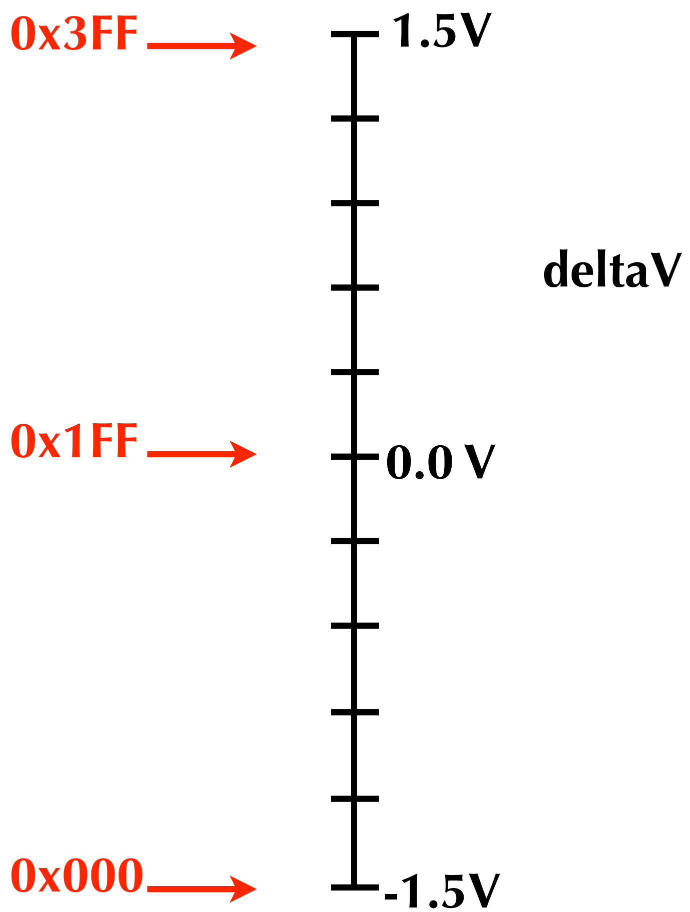
- If $\Delta V = 0$, then the conversion result = 0x1FF
- If $\Delta V > 0$, then the conversion result > 0x1FF (range is 0x1FF–0x3FF)
- If $\Delta V < 0$, then the conversion result < 0x1FF (range is 0–0x1FF)

The differential pairs assign polarities to the analog inputs: the even-numbered input is always positive, and the odd-numbered input is always negative. In order for a valid conversion result to appear, the negative input must be in the range of ± 1.5 V of the positive input. If an analog input is greater than 3 V or less than 0 V (the valid range for analog inputs), the input voltage is clipped, meaning it appears as either 3 V or 0 V, respectively, to the ADC.

Differential Inputs

deltaV = ADCeven - ADCodd

Used for common voltage rejection



ADC Initialization

Enable ADC0

Config. ADC0/Seq0 Prog. Trig.

```
unsigned long ulValue;

SysCtlPeripheralEnable( SYSCTL_PERIPH_ADC0 );

//
// Enable the first sample sequencer to capture the value of channel 0 when
// the processor trigger occurs.
//
ADCSequenceConfigure(ADC0_BASE, 0, ADC_TRIGGER_PROCESSOR, 0);
ADCSequenceStepConfigure(ADC0_BASE, 0, 0,
                         ADC_CTL_IE | ADC_CTL_END | ADC_CTL_CH0);
ADCSequenceEnable(ADC0_BASE, 0);


void ADCSequenceConfigure(unsigned long ulBase,
                          unsigned long ulSequenceNum,
                          unsigned long ulTrigger,
                          unsigned long ulPriority)

void ADCSequenceStepConfigure(unsigned long ulBase,
                              unsigned long ulSequenceNum,
                              unsigned long ulStep,
                              unsigned long ulConfig)
```

ADC Step Configuration

Config. ADC0/Seq0

Step Index

```
unsigned long ulValue;

SysCtlPeripheralEnable( SYSCTL_PERIPH_ADC0 );

//
// Enable the first sample sequencer to capture the value of channel 0 when
// the processor trigger occurs.
//
ADCSequenceConfigure(ADC0_BASE, 0, ADC_TRIGGER_PROCESSOR, 0);
ADCSequenceStepConfigure(ADC0_BASE, 0, 0,
                         ADC_CTL_IE | ADC_CTL_END | ADC_CTL_CH0);
ADCSequenceEnable(ADC0_BASE, 0);
```

Int. Enable

Last Step

ADC0/Ch0

```
void ADCSequenceConfigure(unsigned long ulBase,
                          unsigned long ulSequenceNum,
                          unsigned long ulTrigger,
                          unsigned long ulPriority)

void ADCSequenceStepConfigure(unsigned long ulBase,
                              unsigned long ulSequenceNum,
                              unsigned long ulStep,
                              unsigned long ulConfig)
```

Information and Telecommunication Technology Center

THE UNIVERSITY OF KANSAS

ADC Trigger, Wait, Read Result

```
//
// Trigger the sample sequence.
//
ADCProcessorTrigger(ADC0_BASE, 0);

//
// Wait until the sample sequence has completed.
//
while(!ADCIntStatus(ADC0_BASE, 0, false))
{
}

//
// Read the value from the ADC.
//
ADCSequenceDataGet(ADC0_BASE, 0, &ulValue);
```

Start

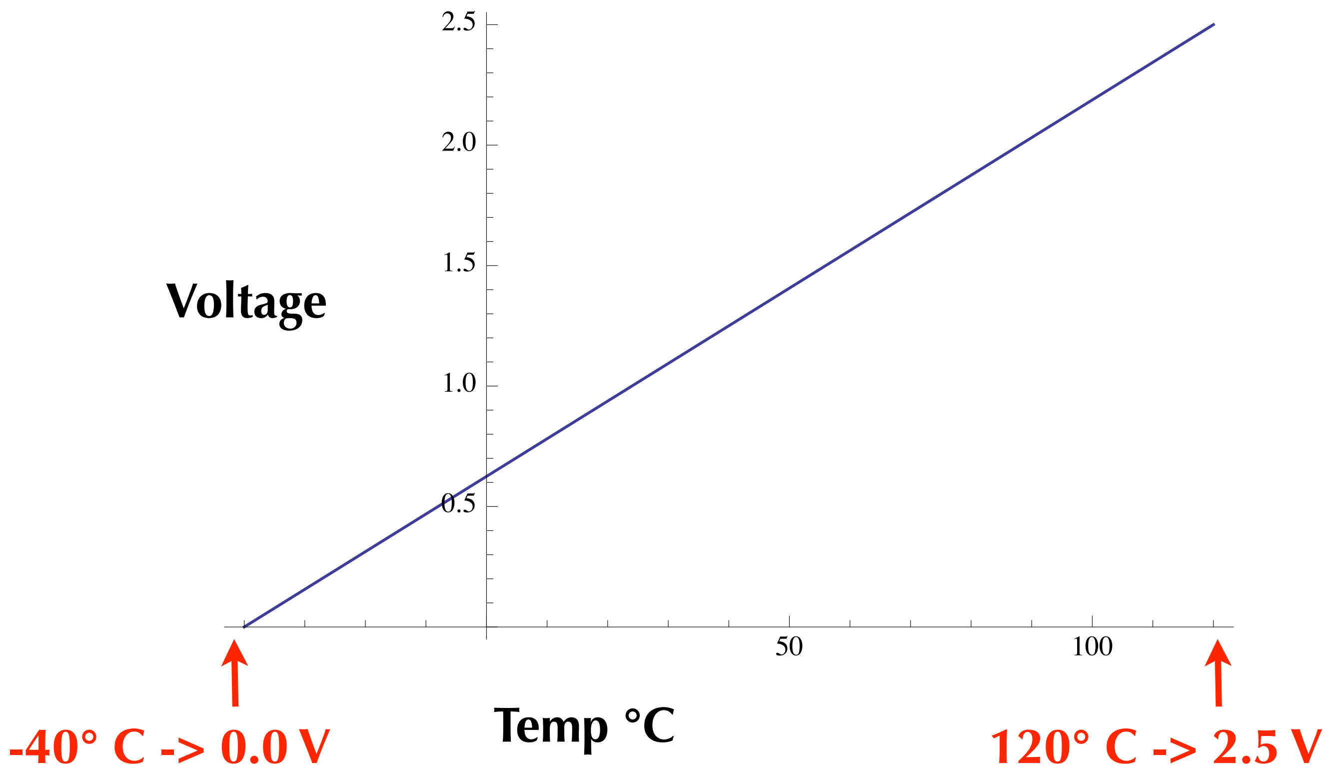
Wait

Read Date

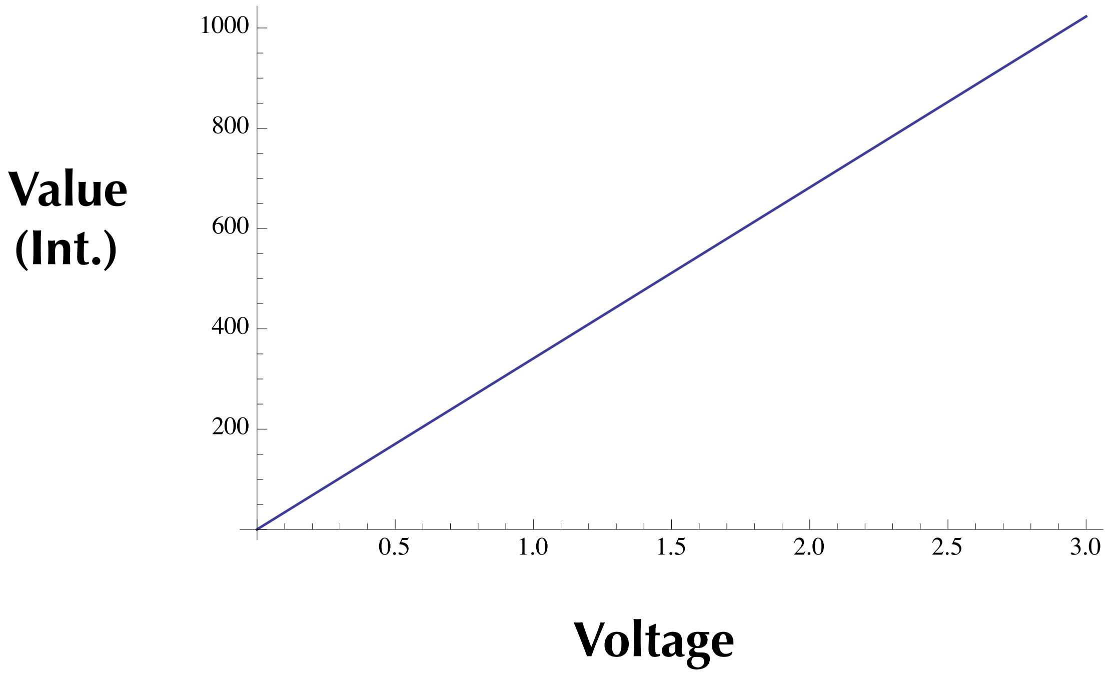
Sensor Value Mapping

- Sensor is electronic device
  - Thermistor, Photo-Transistor, Microphone, ...
- Each sensor has a physical measurement range
  - E.g. -40° C to 120°C
- Each sensor maps measurement range a voltage range based on sensor and electronic circuit
  - E.g. Map -40° C to 120°C to 0.0 V to 2.5 V
- Micro-controller measures voltage
  - E.g. Voltage between 0.0 V and 3.0 V on LM3S1968
- Program has to interpret voltage measurement

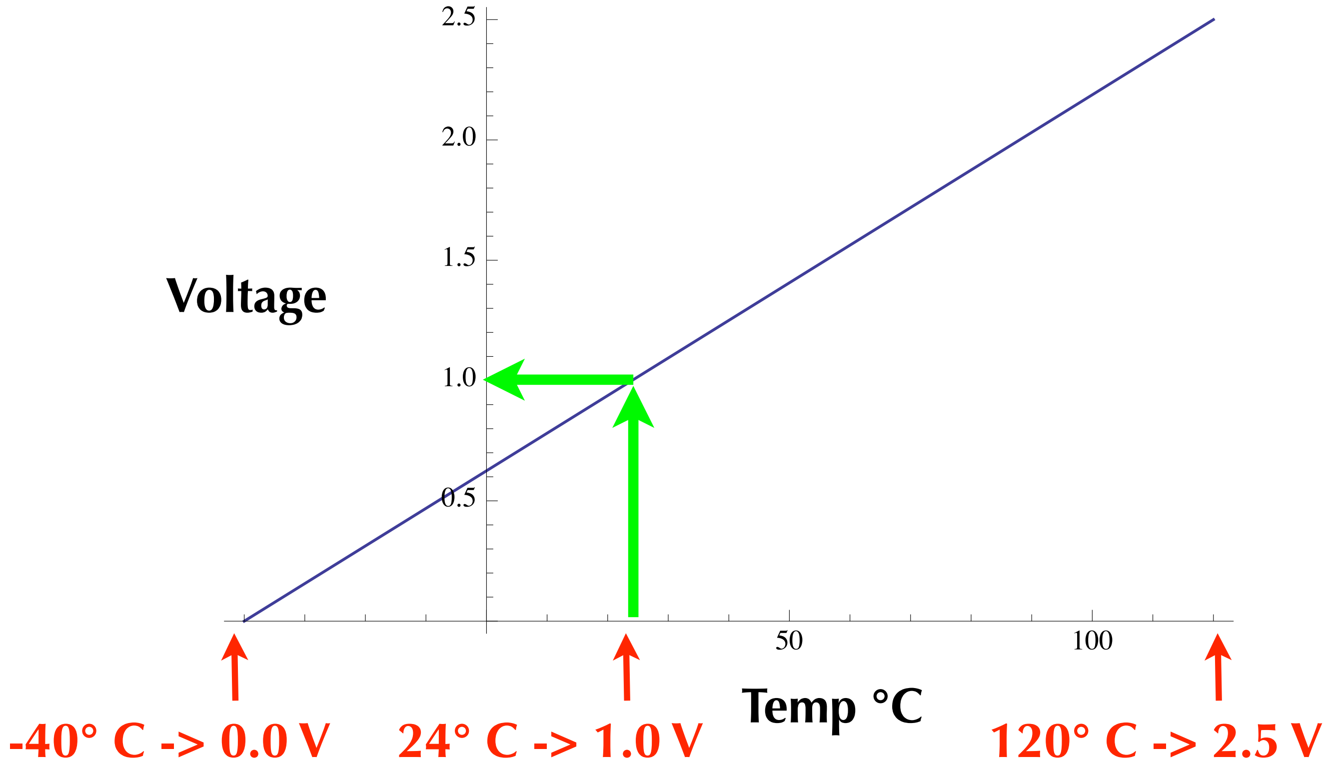
Sensor to Voltage Mapping



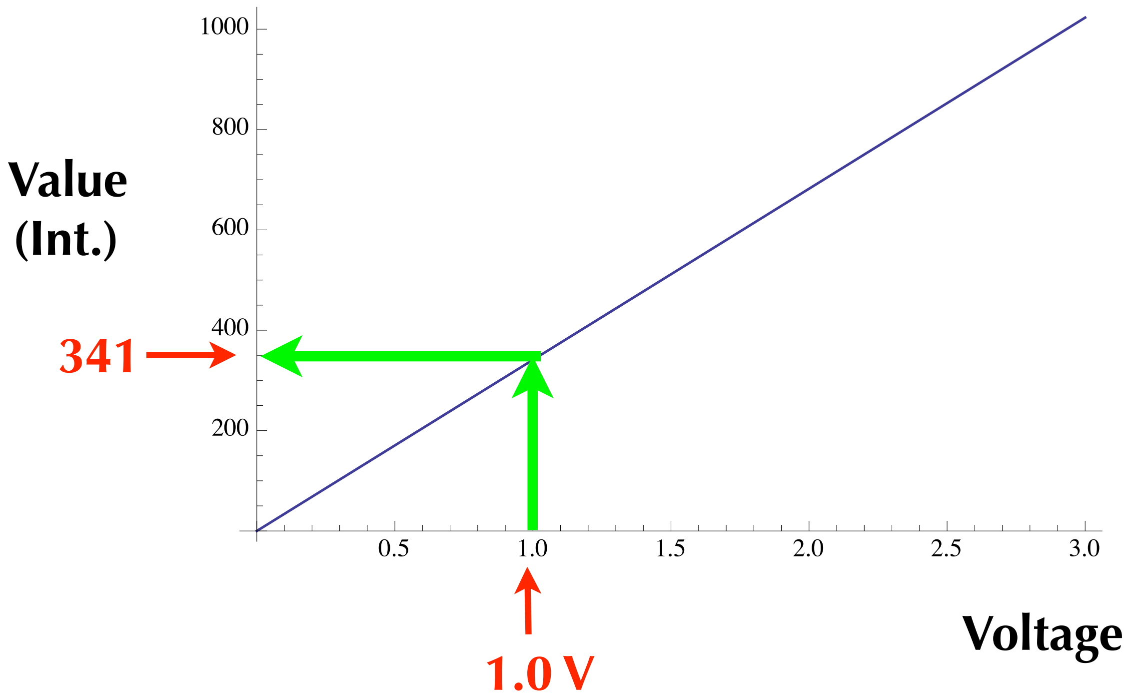
ADC Voltage to Value Mapping



Sensor to Voltage Mapping

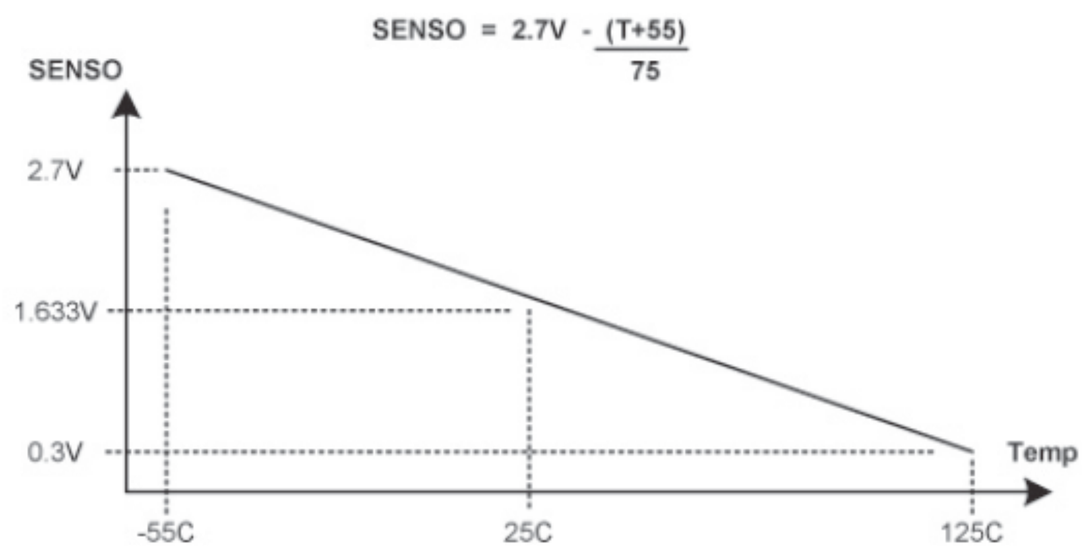


ADC Voltage to Value Mapping



LM3S1968 Temperature Sensor

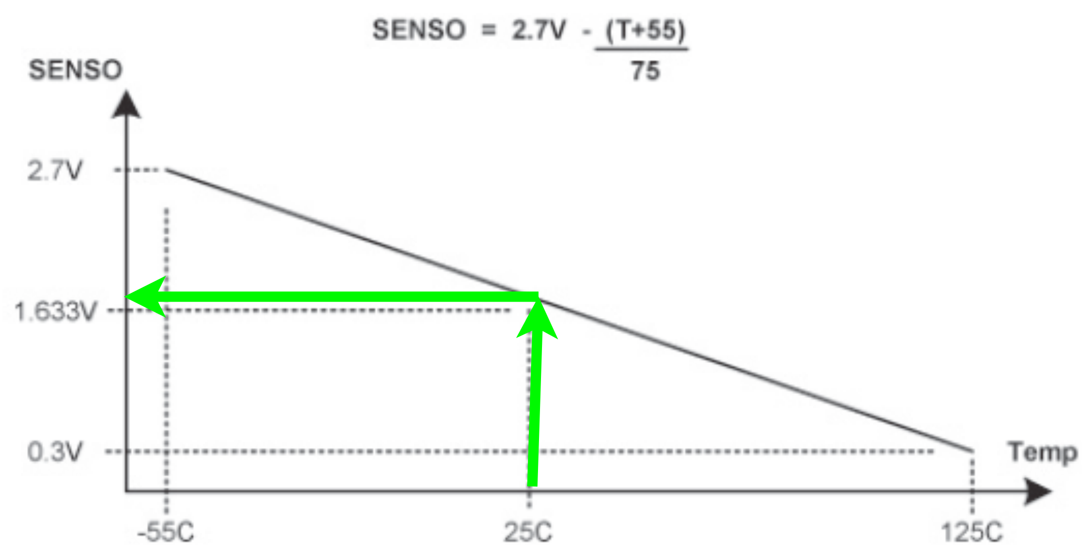
Figure 11-5. Internal Temperature Sensor Characteristic



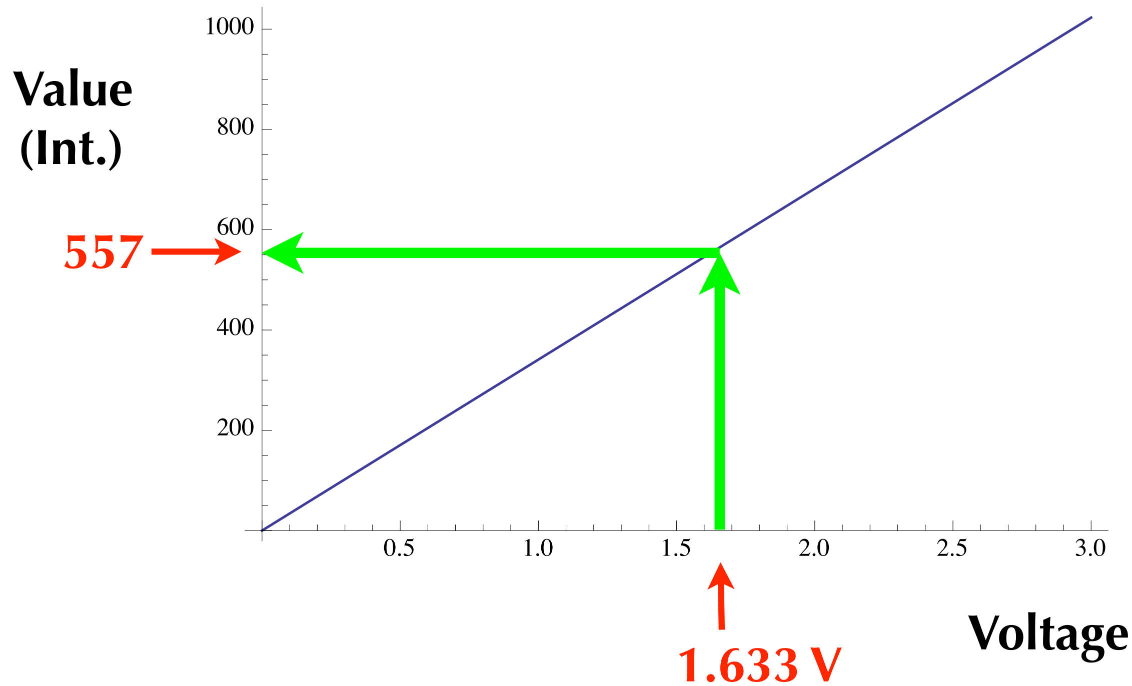
LM3S1968 Datasheet, pg. 400

LM3S1968 Temperature Sensor

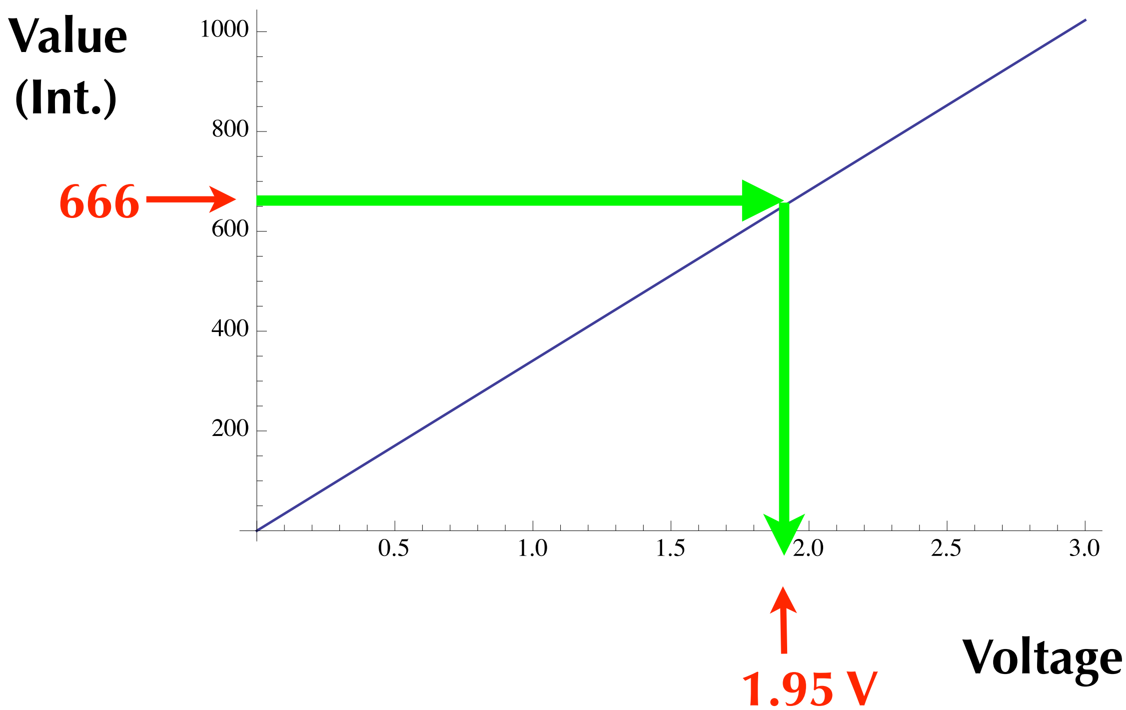
Figure 11-5. Internal Temperature Sensor Characteristic



ADC Voltage to Value Mapping



ADC Voltage to Value Mapping



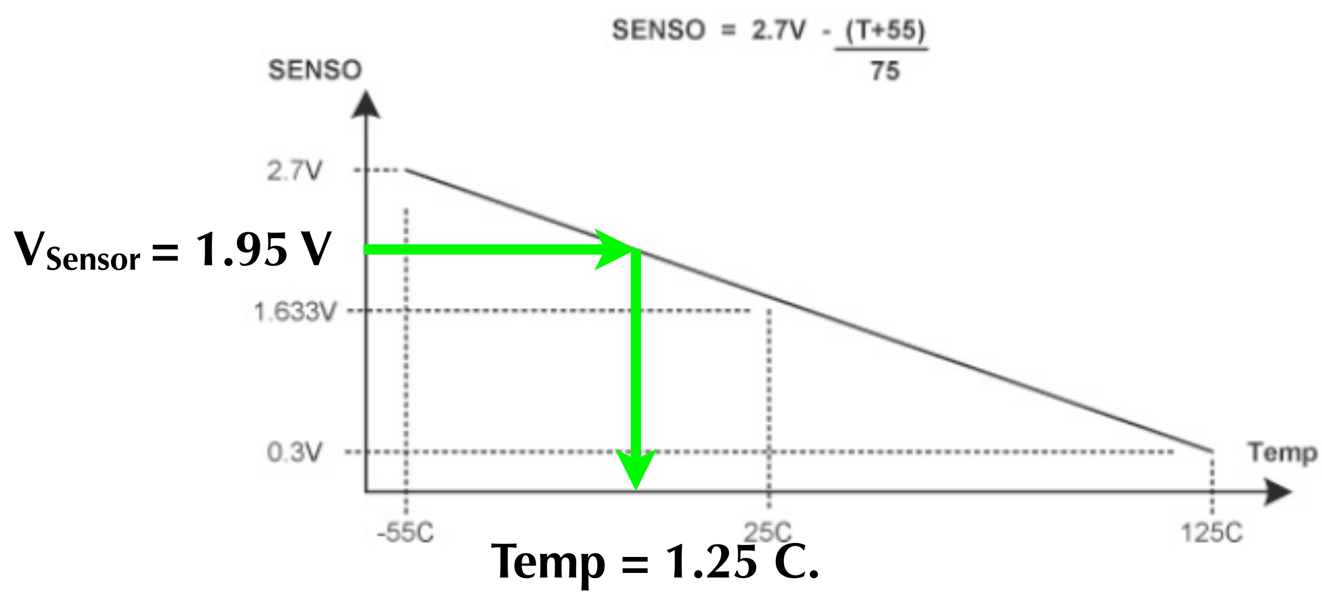
666/1024 * 3.0 V = 0.6504 * 3.0 = 1.95 V = V_{Sensor}

Information and Telecommunication Technology Center

THE UNIVERSITY OF KANSAS

LM3S1968 Temperature Sensor

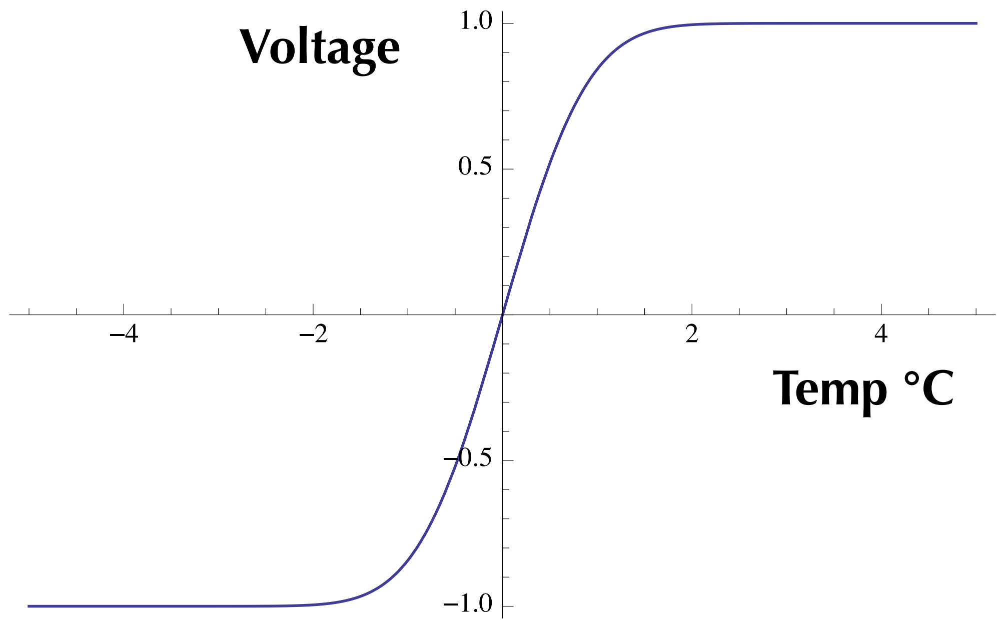
Figure 11-5. Internal Temperature Sensor Characteristic



V_{Sensor} = 1.95 V

Temp = - ((V_{Sensor} - 2.7 V) * 75) - 55 = 1.25 C

Non-Linear Sensors



BMP 180 -- Reading Raw Values

read uncompensated temperature value
write 0x2E into reg 0xF4, wait 4.5ms
read reg 0xF6 (MSB), 0xF7 (LSB)
UT = MSB << 8 + LSB

↓

read uncompensated pressure value
write 0x34+(oss<<6) into reg 0xF4, wait
read reg 0xF6 (MSB), 0xF7 (LSB), 0xF8 (XLSB)
UP = (MSB<<16 + LSB<<8 + XLSB) >> (8-oss)

		bmp180_get_ut	
UT =	27898		long
oss =	0		
= oversampling_setting			short (0 .. 3)
(ultra low power mode)		bmp180_get_up	
UP =	23843		long

BMP 180 -- Calculate Temperature

calculate true temperature
$X1 = (UT - AC6) * AC5 / 2^{15}$
$X2 = MC * 2^{11} / (X1 + MD)$
$B5 = X1 + X2$
$T = (B5 + 8) / 2^4$

		bmp180_get_temperature	
X1 =	4743		long
X2 =	-2344		long
B5 =	2399		long
T =	**150**	***temp in 0.1°C***	long

Information and Telecommunication Technology Center

THE UNIVERSITY OF KANSAS

BMP 180 -- Calculate Pressure

calculate true pressure
$B6 = B5 - 4000$
$X1 = (B2 * (B6 * B6 / 2^{12})) / 2^{11}$
$X2 = AC2 * B6 / 2^{11}$
$X3 = X1 + X2$
$B3 = (((AC1*4+X3) << oss) + 2) / 4$
$X1 = AC3 * B6 / 2^{13}$
$X2 = (B1 * (B6 * B6 / 2^{12})) / 2^{16}$
$X3 = ((X1 + X2) + 2) / 2^{2}$
$B4 = AC4 * (\text{unsigend long})(X3 + 32768) / 2^{15}$
$B7 = ((\text{unsigned long})UP - B3) * (50000 >> oss)$
if $(B7 < 0x80000000)$ { $p = (B7 * 2) / B4$ }
else { $p = (B7 / B4) * 2$ }
$X1 = (p / 2^{8}) * (p / 2^{8})$
$X1 = (X1 * 3038) / 2^{16}$
$X2 = (-7357 * p) / 2^{16}$
$p = p + (X1 + X2 + 3791) / 2^{4}$

		BMP180_calpressure	
B6 =	-1601		long
X1 =	1		long
X2 =	56		long
X3 =	57		long
B3 =	422		long
X1 =	2810		long
X2 =	59		long
X3 =	717		long
B4 =	33457		unsigned long
B7 =	1171050000		unsigned long
p =	70003		long
			long
X1 =	74529		long
X1 =	3454		long
X2 =	-7859		long
p =	**69964**	***press. in Pa***	long

Information and Telecommunication Technology Center

THE UNIVERSITY OF KANSAS

BMP 180 -- Altitude

$$\text{altitude} = 44330 * \left(1 - \left(\frac{p}{p_0}\right)^{\frac{1}{5.255}}\right)$$

Thus, a pressure change of Δp = 1hPa corresponds to 8.43m at sea level.

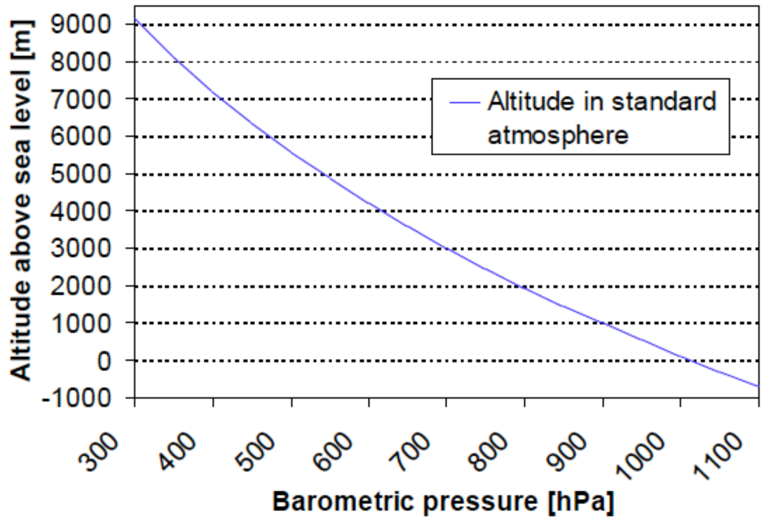


Figure 5: Transfer function: Altitude over sea level – Barometric pressure

Information and Telecommunication Technology Center

THE UNIVERSITY OF KANSAS