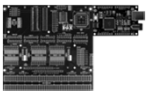
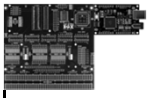


EEL 3744

Menu

- A/D system on the 68HC11/12 & TI DSC F2833
- A/D system on the XMEGA
- A/D Converter Example: EEG
- Analog-to-Digital Conversion
  - >Basic Charge-Redistribution A/D
- Analog-to-Digital Conversion
  - >What should the answers be?
  - >Example of 2-, 4-, … 8-bit conversions

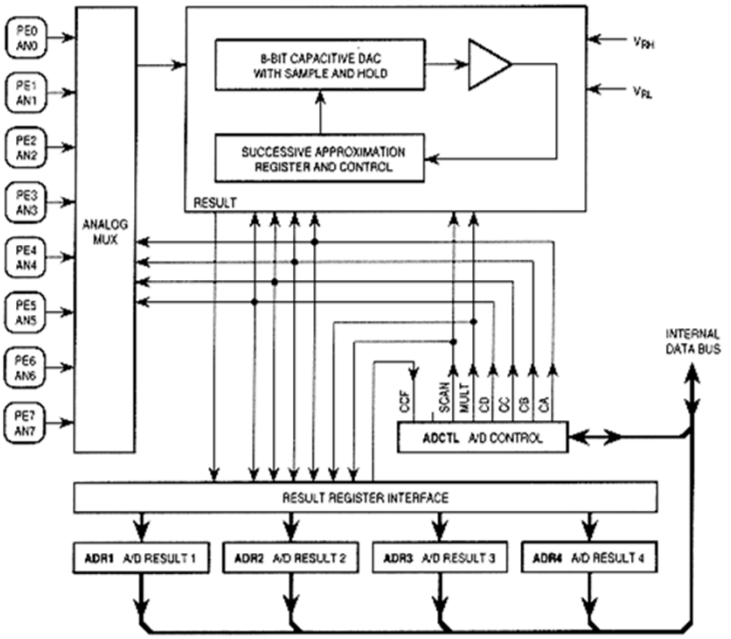
See examples on web-site:
doc8331, doc8032, AD_EEG.asm

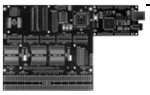


EEL 3744

68HC11
A/D Block Diagram

TD:Fig.10-1

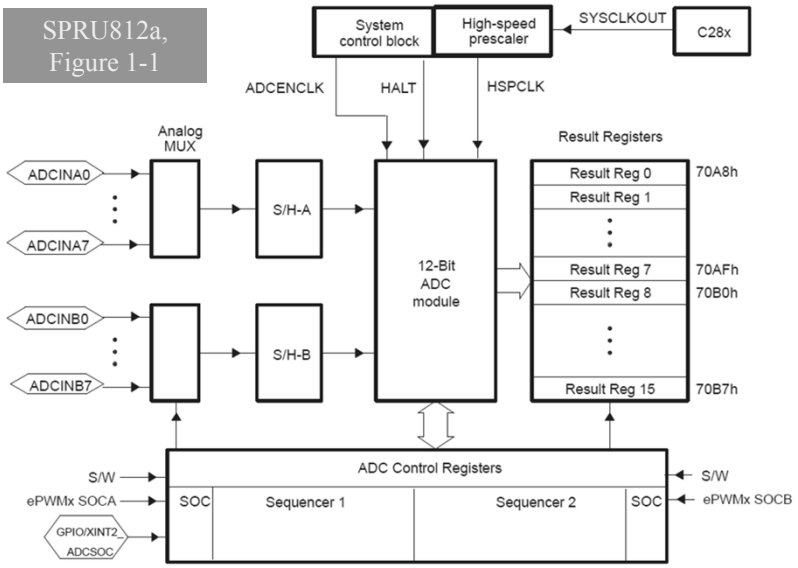


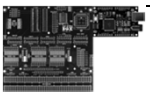


EEL 3744 **DSC** ADC Block Diagram

SPRU812a, Figure 1-1

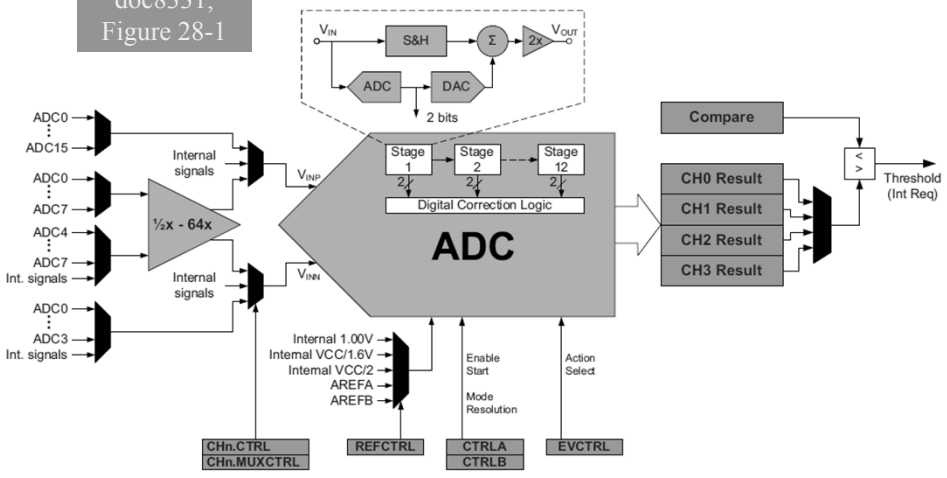
- Two 8-channel 12-bit ADCs on our DSC boards

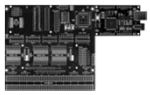




EEL 3744 XMEGA ADC Block Diagram

doc8331, Figure 28-1



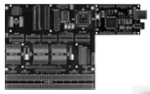


EEL 3744

XMEGA ADC Features

See doc8331, Section 28.1

- 12-bit resolution
- Up to 2Msamples/sec
  - > 2 inputs sampled simultaneously
  - > 4 inputs sampled within 1.5µs (667kHz)
- Differential or single-ended input
  - > Differential inputs with or without gain
    - – Gains: ½ ×, 1×, 2×, 4×, 8×, 16×, 32x, 64×
- Single scan or continuous scans
- Signed or unsigned results
- Internal and external reference options
- Optional event triggered conversion
- Four conversion channels with individual input control and result register
  - > Enable four parallel configurations and results

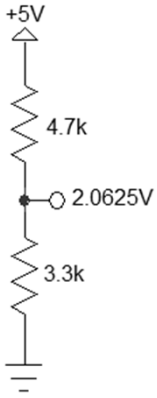


EEL 3744

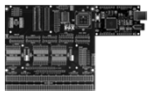
uTinkerer ADC

See uTinkerer Manual, Page 8

- "The uTinkerer, in an effort to reduce external circuitry size and complexity, has built in pre-amplifiers on ADC (port A) inputs **AD0-7**. The pre-amplifiers essentially **remap the range of the ADC from 0 to 2.0625V to** a more flexible **0 to 5V**. The pre-amplifiers are designed to work with the internal ADC voltage reference (VREF) of VCC/1.6V (2.0625V when VCC=3.3V)."
- "It should be noted that even though the pre-amplifier circuit increases functionality, it also is another source for analog conversion error. For details, consult the included sensitivity analysis of the analog pre-amplifiers."
- ADC input pins **AD8-11** are raw inputs. They connect **directly** to the ATxmega128A1U, i.e., with **no** resistor divider circuits.
- If put 5V on top, get 2.0625V at center tap.



$$VREF = \frac{VCC}{1.6} = \frac{3.3}{1.6} = 2.0625\ \text{V} \qquad \frac{5\ \text{V} * 3.3\text{k}}{4.7\text{k} + 3.3\text{k}} = 2.0625\ \text{V}$$

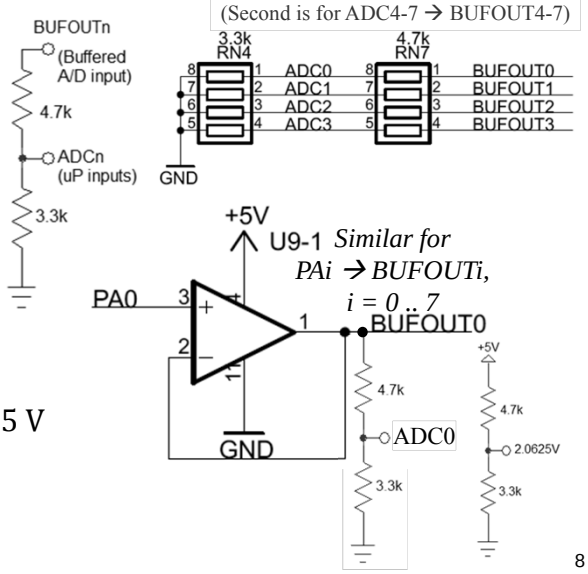


EEL 3744
uTinkerer ADC

See uTinkerer Schematics, Sheet 2

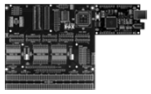
Below is 1 of 2 sets of similar circuits on uTinkerer PCB
(Second is for ADC4-7 → BUFOUT4-7)

- PA0-PA7 are the **external uTinkerer ADC inputs**
  - > Pins are buffered with circuit (on bottom right)
  - > Signals are attenuated with the resistor divider circuit →
  - > PAx pins are on J21
    - – Each PAx can be 0 to 5V
- ADC0-7 (ADCn) go to the XMEGA ADC pins on ports A
  - > Port B ADC pins **different**!



$$VREF = \frac{VCC}{1.6} = \frac{3.3}{1.6} = 2.0625\ \text{V}$$

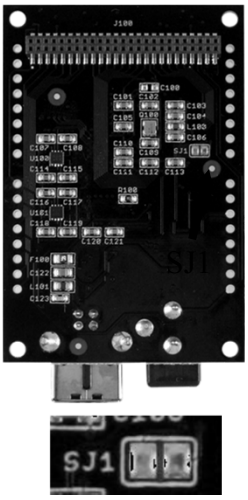
$$\frac{5\ \text{V} * 33\text{k}}{47\text{k} + 33\text{k}} = 2.0625\ \text{V}$$

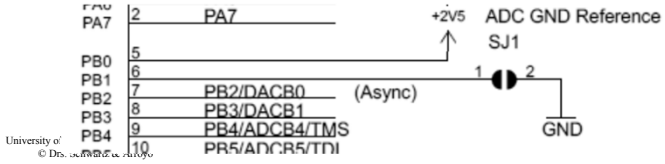


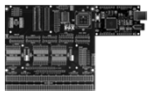
EEL 3744
uPAD ADC

See uPAD Applications Manual: Analog In

- PORTB 0 is the input for the precision 2.5V analog reference
- PORTB 1 is the circuit GND reference input for differential analog measurements
- **If you set the direction register for PORTB pins 0 or 1 to output** (default for all pins is input) **you risk destroying your board!**
  - > **Always use caution when using PORTB of the µPAD's XMEGA!!!**
- The solder jumper SJ1 (on the bottom of the uPAD PCB, highlighted here), connects PB1 to board GND
  - > This GND connection serves as the negative input for differential measurements and the positive input terminal for differential measurements with gain via the XMEGA's PGA (Programmable Gain Array)

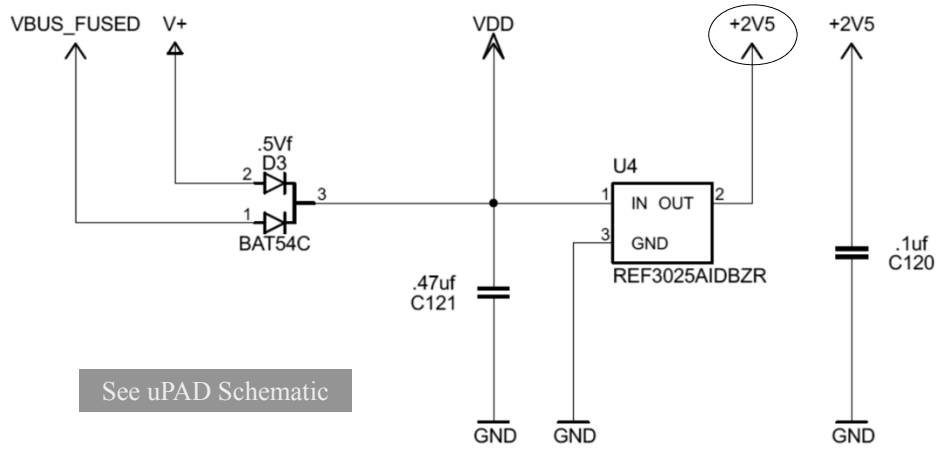




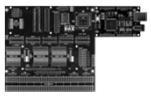


EEL 3744 uPAD ADC

- ADC +2.5V reference schematic



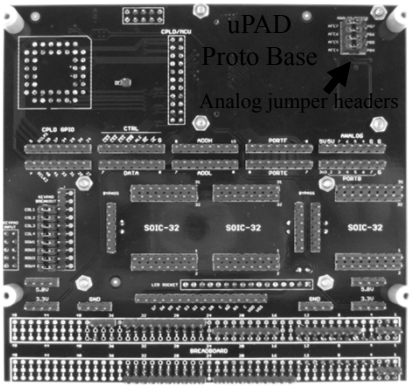
See uPAD Schematic

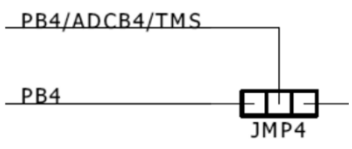


EEL 3744 uPAD ADC

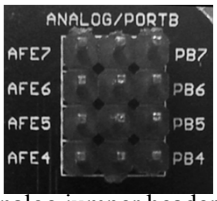
See uPAD Applications Manual: Analog In

- This group of 4 selectors is used to control whether PORTB pins 4-7 are connected to their respective analog front end circuits, or their respective PORTB breakout connections in J9 of the µPAD Proto Base.
- Each pin (PB4-PB7) can be individually configured per its respective selector
- In the figure on the bottom right, all of the PB4-PB7 pins are connected to their respective analog front end circuits (with jumpers connect the middle and left pins).

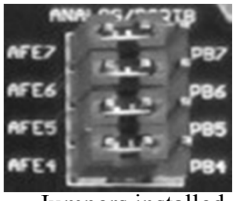




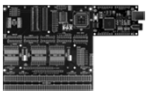
PB5-PB7 are similar



Analog jumper headers



Jumpers installed



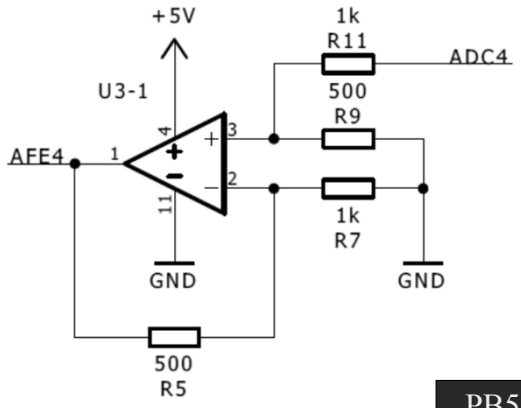
EEL 3744

uPAD ADC Front End

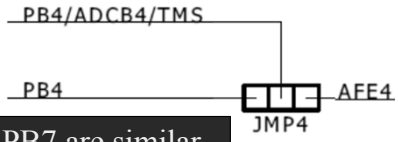
See uPAD Applications Manual: Analog In

- Front end for ADC PB4
- AFE4 = ADC4 / 2
  - > See Application Manual for derivation

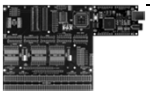
$$AFE4 = \frac{ADC4}{2}$$



- Note that the resistor tolerances are 1%



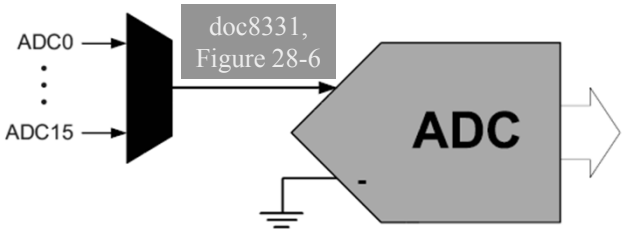
PB5-PB7 are similar



EEL 3744

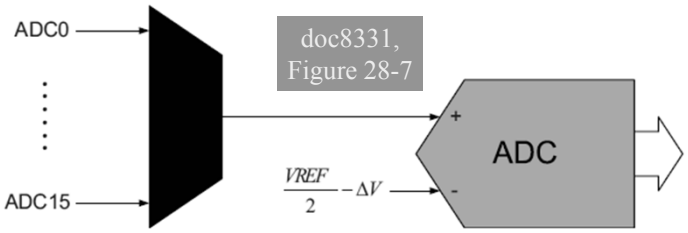
XMEGA ADC: Single-ended measurements

Signed Mode

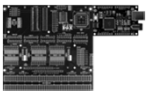


doc8331, Figure 28-6

Unsigned Mode



doc8331, Figure 28-7



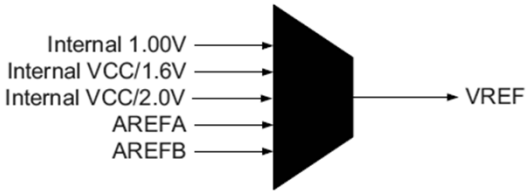
EEL 3744

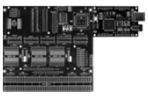
See doc8331, Section 28.5

XMEGA Voltage Reference

- Voltage reference (VREF) for the ADC is set to one of the following
  - >Internal 1.00V
  - >Internal Vcc/1.6V (=2.0625V for Vcc=3.3V)
  - >Internal Vcc/2V
  - >External voltage at AREF pin on PORTA
  - >**External voltage at AREF pin on PORTB**

doc8331, Figure 28-8



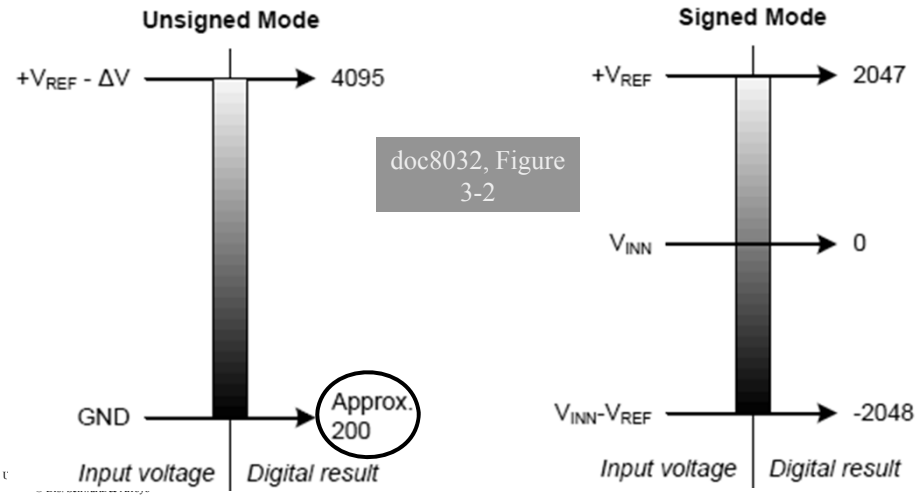


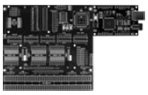
EEL 3744

XMEGA Offset in Unsigned mode

- Note the offset in unsigned mode

doc8032, Figure 3-2



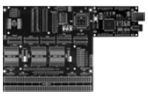


EEL 3744

XMEGA 12-bit or 8-bit Conversion

- ADC can be configure to generate either an **8-bit** or a **12-bit** result
  - > Of course 8-bit results are available faster
- Result registers are 16 bits wide (i.e., two 8-bit registers)
  - > Data can be stored as right adjusted 16-bit values
    - – Right adjusted means the 8 least-significant bits (lsb) are put in the low byte
    - – Left adjusted means the 8 most-significant bits (msb) are put in the high byte
  - > A 12-bit result can be either left or right adjusted
- When in signed mode, the msb represent the sign bit
  - > The sign bit is sign-extend
    - – For 12-bit right adjusted, bit 11 is repeated for bits 15-12
    - – For 8-bit right adjusted, bit 7 is repeated for bits 15-8

See doc8331, Section 28.6



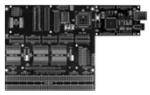
EEL 3744

XMEGA Compare Function

- One 12-bit compare register
  - > Four Analog Comparators
    - – Each of four ADC channels can be set for an interrupt when result is above or below the threshold
  - > Selectable propagation delay versus curent consumption
  - > Selectable hysteresis (none, small, large)
  - > Analog comparator output available on pin
  - > Flexible input selection
    - – All pins on the port
    - – Output from the DAC
    - – Bandgap reference voltage
    - – A 64-level programmable voltage scaler of the internal VCC voltage
  - > Interrupt and event generation on:
    - – Rising edge, Falling edge, Toggle
  - > Window function interrupt and event generation on:
    - – Signal above window, Signal inside window, Signal below window
  - > Constant current source with configurable output pin selection

See doc8331, Section 28.7, 30

See doc8385, Section 31



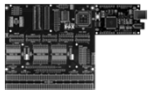
EEL 3744

See doc8331, Section 30

XMEGA Compare Function

See doc8385, Section 31.2

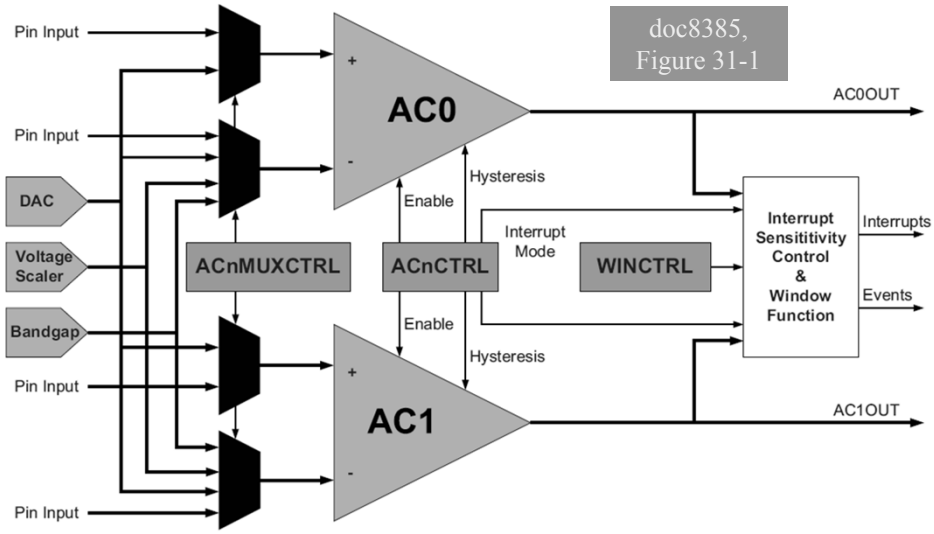
- One 12-bit compare register
  - > Each of four ADC channels can be set for an interrupt when result is above or below the threshold
  - > The input selection includes analog port pins, several internal signals, and a 64-level programmable voltage scaler
    - – The analog comparator output state can also be output on a pin for use by external devices
  - > The analog comparators are always grouped in pairs on each port, called analog comparator 0 (AC0) and analog comparator 1 (AC1)
    - – They have identical behavior, but separate control registers
    - – Used as pair, they can be set in window mode to compare a signal to a voltage range instead of a voltage level
    - – PORTA and PORTB each has one AC pair, called ACA and ACB, respectively

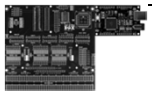


EEL 3744

XMEGA Analog Comparator

doc8385, Figure 31-1

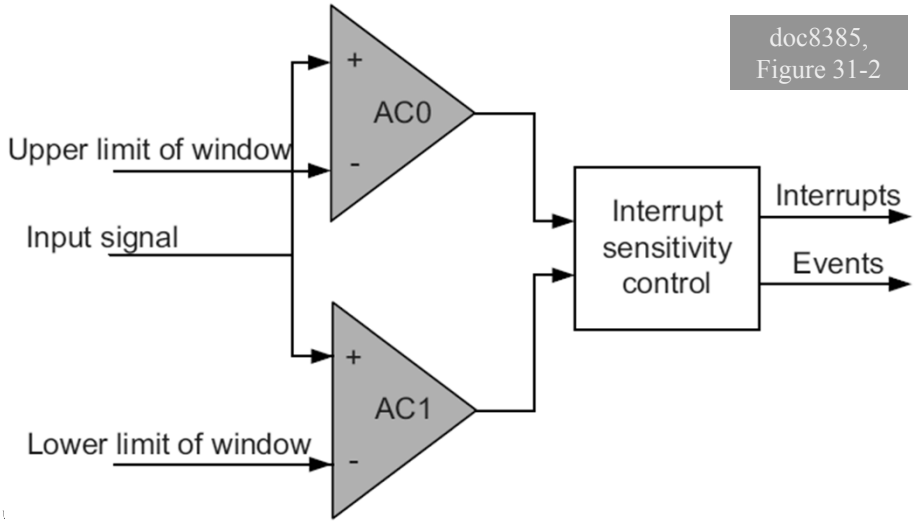




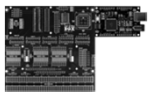
EEL 3744

Analog Comparator Window Function

doc8385, Figure 31-2



20

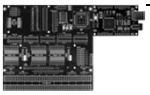


EEL 3744

See doc8331, Section 28.8

XMEGA Analog: Starting a Conversion

- Starting a conversion
  - > Write to the start conversion bit for one or more channels
  - > Use the event system to start one or several conversions
  - > If multiple start conversion bits are written, the scan starts from the lowest channel number
  - > *28.8.1 Input Source Scan:*
    - *For ADC Channel 0 it is possible to select a range of consecutive input sources that is automatically scanned and measured when a conversion is started*
    - *This is done by setting the first (lowest) positive ADC channel input using the MUX control register, and a number of consecutive positive input sources*
    - *When a conversion is started, the first selected input source is measured and converted, then the positive input source selection is incremented after each conversion until it reaches the specified number of sources to scan*

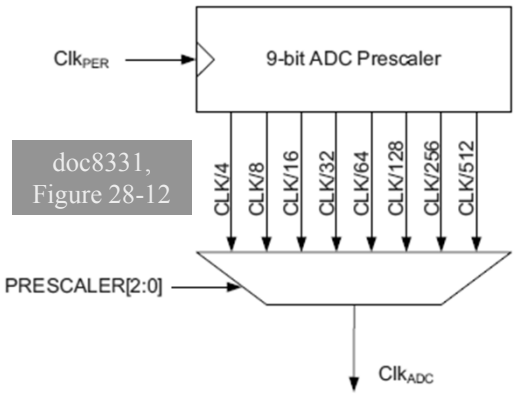


EEL 3744 XMEGA ADC Clock and Conversion Timing

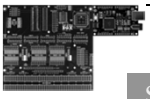
See doc8331, Section 28.9

- In below formula
  - > RESOLUTION = 8 or 12
  - > GAIN=0 (no gain) or 1 (gain)
  - > f_{ADC} = sample rate
- The ADC clock rate is the limiting factor, **NOT** the propagation delay (due to the pipeline)
- The msb (most-significant bit) is converted first, the rest of the bits are converted in the
  - > next 3 ADC clock cycles for 8-bit
  - > next 5 ADC clock cycles for 12-bit
  - > Converting 1 bit takes ½ T_{ADC}
  - > Interrupt flag is set after result register is loaded

doc8331, Figure 28-12



$$\text{Propagation Delay} = \frac{1 + \frac{RESOLUTION}{2} + GAIN}{f_{ADC}}$$

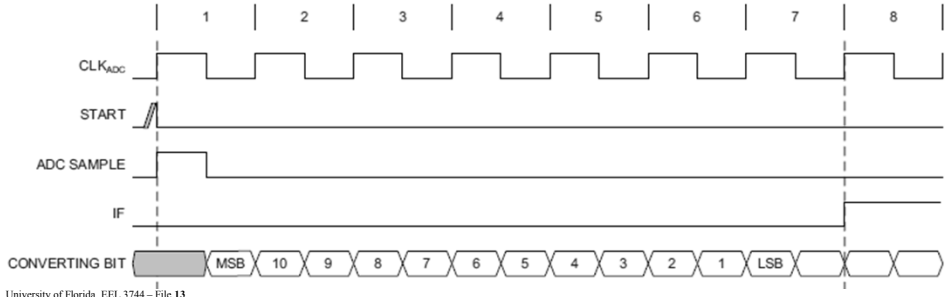


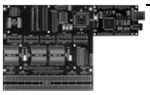
EEL 3744 XMEGA ADC Timing (Single Conversion, no Gain)

See doc8331, Section 28.9.1

- The writing of the start conversion bit, or the event triggering the conversion (START), must occur at least one peripheral clock cycle before the ADC clock cycle on which the conversion starts (indicated with the grey slope of the START trigger)
- The input source is sampled in the first half of the first cycle

doc8331, Figure 28-13

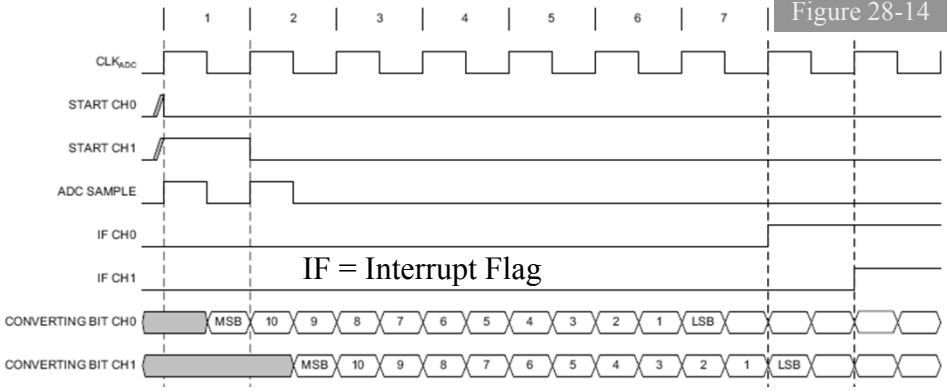


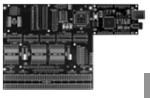


EEL 3744 ADC Timing: Single Conversion, 2 Channels

- The pipelined design enables the second conversion to start on the next ADC clock cycle after the first conversion has started.
- Both conversions take place at the same time, but the conversion on ADC channel 1 (CH1) does not start until the ADC samples and performs conversion on the msb on channel 0 (CH0)

doc8331, Figure 28-14





EEL 3744 XMEGA ADC CTRLA – Control register A

See doc8331, Section 28.16.1

- **DMASEL: DMA Request Selection**
  - > Can allow one DMA channel to serve more than one ADC channel

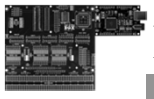
doc8331, Table 28-1

DMASEL[1..0]	Group Config	Description
00	OFF	No combined DMA request
01	CH01	Common request for ADC channels 0 & 1
10	CH012	Common request for ADC channels 0, 1 & 2
11	CH0123	Common request for ADC channels 0, 1, 2 & 3

- **CHSTART[3:0]: Channel Start Single Conversion**
  - > Setting bits will **start a conversion** on the corresponding ADC channel

Bit	7	6	5	4	3	2	1	0
+0x00	DMASEL[1:0]		CHSTART[3:0]				FLUSH	ENABLE
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

ADCn_CTRLA, n=A,B



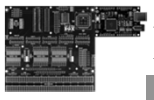
EEL 3744 XMEGA ADC CTRLA – Control register A

See doc8331, Section 28.16.1

- **CHSTART[3:0]: Channel Start Single Conversion**
  - > Setting bits will **start a conversion** on the corresponding ADC channel; if several started at same time, lowest channel will start 1st
- **FLUSH: Pipeline Flush**
  - > Set to flush the ADC pipeline
    - – ADC clock will restart on next peripheral clock edge & resume where left off
    - – Pending conversion will enter the ADC pipeline and complete
  - > All conversions are aborted and lost
- **ENABLE: Enable**
  - > Set this bit to enable the ADC

Bit	7	6	5	4	3	2	1	0
+0x00	DMASEL[1:0]		CHSTART[3:0]				FLUSH	ENABLE
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

ADCn_CTRLA, n=A,B



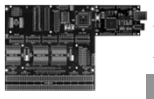
EEL 3744 ADC CTRLB – ADC Control register B

See doc8331, Section 28.16.2

- **IMPMODE: Gain Stage Impedance Mode**
  - > 0 = high impedance sources; 1 = low impedance sources
- **CURRLIMIT[1:0]: Current Limitation**
  - > Control current consumption by reducing the max ADC sample rate
- **CONVMODE: Conversion Mode**
  - > 0 = unsigned mode; 1 = signed
- **FREERUN: Free Running Mode**
  - > 0 = single scan
  - > 1 = free running mode
    - – ADC channels defined in EVCTRL register are swept repeatedly

Bit	7	6	5	4	3	2	1	0
+0x01	IMPMODE	CURRLIMIT[1:0]		CONVMODE	FREERUN	RESOLUTION[1:0]		–
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Initial Value	0	0	0	0	0	0	0	0

ADCn_CTRLB, n=A,B



EEL 3744

See doc8331, Section 28-16.2

ADC CTRLB – ADC Control register B

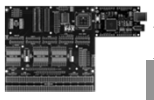
- **FREERUN: Free Running Mode**
  - > 0 = single scan
  - > 1 = free running mode
    - – ADC channels defined in EVCTRL register are swept repeatedly
- **RESOLUTION[1:0]: Conversion Result Resolution**

doc8331, Table 28-4

RESOLUTION[1..0]	Group Config	Description
00	12-bit	12-bit result, right justified
01	-	Reserved
10	8-bit	8-bit result, right justified
11	Left 12-bit	12-bit result, left justified

Bit	7	6	5	4	3	2	1	0
+0x01	IMPMODE	CURRLIMIT[1:0]		CONVMODE	FREERUN	RESOLUTION[1:0]		–
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Initial Value	0	0	0	0	0	0	0	0

ADCn_CTRLB, n=A,B



EEL 3744

See doc8331, Section 28.16.3

ADC REFCTRL – Reference Control register

- **REFSEL[2:0]: Reference Selection**
  - > Selects the reference for the ADC

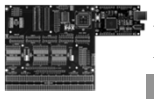
doc8331, Table 28-5

REFSEL[1..0]	Group Config	Description
000	INT1V	10/11 of bandgap (1.0V)
001	INTVCC	Vcc/1.6
010	AREFA	External ref from AREF pin or PORT A
011	**AREFB**	**External ref from AREF pin or PORT B**
100	INVCC2	Vcc/2
101-111	-	Reserved

- **BANDGAP: Bandgap Enable**
- **TEMPREF: Temperature Reference Enable**

Bit	7	6	5	4	3	2	1	0
+0x02	–	REFSEL[2:0]			–	–	BANDGAP	TEMPREF
Read/Write	R	R/W	R/W	R/W	R	R	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

ADCn_REFCTRL, n=A,B



EEL 3744 ADC EVCTRL – Event Control register

See doc8331, Section 28.16.4

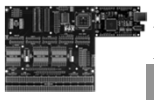
- **SWEEP[1:0]: Channel Sweep**
  - > Control which ADC channels are included in a channel sweep triggered by the event or when in free running mode

doc8331, Table 28-6

SWEEP[1..0]	Group Config	Active ADC channels for channel sweep
00	0	Only ADC channel 0
01	01	ADC channels 0 and 1
10	012	ADC channels 0, 1, and 2
11	0123	ADC channels 0, 1, 2, and 3

Bit	7	6	5	4	3	2	1	0
+0x03	SWEEP[1:0]		EVSEL[2:0]			EVACT[2:0]		
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

ADCn_EVCTRL, n=A,B



EEL 3744 ADC EVCTRL – Event Control register

See doc8331, Section 28.16.4

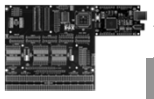
- **EVSEL[2:0]: Event Channel Input Select**
  - > Selects which event channel trigger which ADC channel (ch)
  - > Event ch with lowest # will trigger ADC ch0, the next ADC ch1, …

doc8331, Table 28-7

EVSEL[1..0]	Group Config	Selected event lines
000	0123	Event channel 0, 1, 2, and 3 as selected inputs
001	1234	Event channel 1, 2, 3, and 4 as selected inputs
010	2345	Event channel 2, 3. 4, and 5 as selected inputs
011	3456	Event channel 3, 4, 5, and 6 as selected inputs
100	4567	Event channel 4, 5, 6, and 7 as selected inputs
101	567	Event channel 5, 6, and 7 as selected inputs
110	67	Event channel 6, and 7 as selected inputs
111	7	Event channel 7 as selected input

Bit	7	6	5	4	3	2	1	0
+0x03	SWEEP[1:0]		EVSEL[2:0]			EVACT[2:0]		
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

ADCn_EVCTRL, n=A,B



EEL 3744

See doc8331, Section 28.16.4

ADC EVCTRL – Event Control register

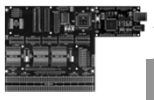
- **EVACT[2:0]: Event Mode**
  - > Selects and limits how many of the selected event input channels are used
  - > Further limits the ADC triggers

doc8331, Table 28-8

EVACT[1..0]	Group Config	Selected event lines
000	None	No event inputs
001	CH0	Event ch with lowest # defined by EVSEL triggers conv on ADC ch 0
010	CH01	" 2 lowest # " on ADC chs 0 & 1
011	CH012	" 3 lowest # " on ADC chs 0, 1, & 2
100	CH0123	Event ch defined by EVSEL triggers conversion on ADC chs 0-3
101	SWEEP	One sweep of all ADC chs defined by SWEEP on incoming event channel with the lowest # defined by EVSEL
110	SYNC SWEEP	1 sweep of all active ADC chs defined by SWEEP on incoming event ch with lowest # defined by EVSE. ADC flushed & restarted.
111	-	Reserved

ADCn_EVCTRL, n=A,B

Bit	7	6	5	4	3	2	1	0
+0x03	SWEEP[1:0]		EVSEL[2:0]			EVACT[2:0]		
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0



EEL 3744

See doc8331, Section 28.16.5

ADC PRESCALER – Clock Prescaler register

- **PRESCALER[2:0]: Prescaler Configuration**
  - > Defines the ADC clock relative to the peripheral clock

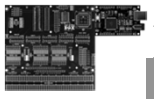
doc8331, Table 28-9

PRESCALER[2..0]	Group Config	Peripheral clock division factor
000	DIV4	4
001	DIV8	8
010	DIV16	16
011	DIV32	32
100	DIV64	64
101	DIV128	128
110	DIV256	256
111	DIV512	512

$$f_{ADC} = \frac{f_{PER}}{2^{(PreScaler+2)}}$$

Bit	7	6	5	4	3	2	1	0
+0x04	–	–	–	–	–	PRESCALER[2:0]		
Read/Write	R	R	R	R	R	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

ADCn_PRESCALER, n=A,B



EEL 3744

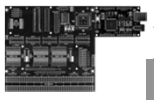
See doc8331, Section 28.16.6

ADC INTFLAGS – Interrupt Flag register

- **CH[3:0]IF: Interrupt Flags**
  - > Set when the ADC conversion is complete for the corresponding ADC channel
  - > If an ADC channel is configured for compare mode, the corresponding flag will be set if the compare condition is met
  - > CHnIF is automatically **cleared** when the ADC channel n **interrupt vector is executed**
  - > Writing a one to the flag's bit location will clear the flag

Bit	7	6	5	4	3	2	1	0
+0x06	–	–	–	–	CH[3:0]IF			
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

ADCn_CHx_INTFLAGS, x=0,1,2,3, n=A,B



EEL 3744

See doc8331, Section 28.17.1

ADC CTRL – Channel Control register

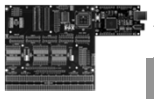
- **START: START Conversion on Channel**
  - > Setting this bit will start a conversion on the channel
  - > The bit is cleared by hardware when the conversion has started
  - > Setting this bit when it already is set will have no effect
  - > Writing or reading this bit is equivalent to writing the CH[3:0]START bits in CTRLA (Control register A)
- **GAIN[2:0]: Gain Factor**
  - > These bits define the gain factor for the ADC gain stage

doc8331, Table 28-10

Gain[2..0]	Group Config	Gain Factor
000	1X	1x
001	2X	2x
010	4X	4x
011	8X	8x
100	16X	16x
101	32X	32x
110	64X	64x
111	DIV2	½ x

Bit	7	6	5	4	3	2	1	0
+0x00	START	–	–	GAIN[2:0]			INPUTMODE[1:0]	
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

ADCn_CHx_CTRL, x=0,1,2,3, n=A,B



EEL 3744

See doc8331, Section 28.17.1

ADC CTRL – Channel Control register

- **INPUTMODE[1:0]: Channel Input Mode**
  - > These bits define the channel mode
  - > Changing input mode will corrupt any data in the pipeline

Channel input modes, CONVMODE=0 (unsigned mode)

InputMode [1..0]	Group Config	Description
00	Internal	Internal positive input signal
01	Single Ended	Single-ended positive input signal
10	-	Reserved
11	-	Reserved

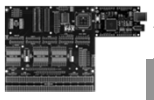
Channel input modes, CONVMODE=1 (signed mode)

InputMode [1..0]	Group Config	Description
00	Internal	Internal positive input signal
01	Single Ended	Single-ended positive input signal
10	Diff	Differential input signal
11	Diff W/ Gain	Differential input signal w/ gain

doc8331, Table 28-11,12

Bit	7	6	5	4	3	2	1	0
+0x00	START	–	–	GAIN[2:0]			INPUTMODE[1:0]	
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

ADCn_CHx_CTRL, x=0,1,2,3, n=A,B



EEL 3744

See doc8331, Section 28.17.2

ADC MUXCTRL – ADC Channel MUX Control registers

- **MUXPOS[3:0]: MUX Selection on Positive ADC Input**
  - > These bits define the MUX selection for the positive ADC input selection

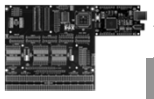
INPUTMODE[1:0] = 01 (single-ended) [see manual for others]

MUXPOS [3..0]	Group Config	Description
0000	PIN0	ADC0 pin
0001	PIN1	ADC1 pin
0010	PIN2	ADC2 pin
0011	PIN3	ADC3 pin
…	PINx	ADCx pin
1111	PIN15	ADC15 pin

doc8331, Table 28-14

Bit	7	6	5	4	3	2	1	0
+0x01	–	MUXPOS[3:0]				MUXNEG[2:0]		
Read/Write	R	R/W	R/W	R/W	R/W	R	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

ADCn_CHx_MUXCTRL, x=0,1,2,3, n=A,B



EEL 3744

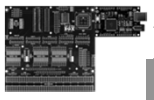
See doc8331, Section 28.17.2

ADC MUXCTRL – ADC Channel MUX Control registers

- **MUXNEG[2:0]: MUX Selection on Negative ADC Input**
  - > These bits define the MUX selection for the negative ADC input when differential measurements are done
  - > For **internal or single-ended** measurements, these bits are **not used**

Bit	7	6	5	4	3	2	1	0
+0x01	–	MUXPOS[3:0]				MUXNEG[2:0]		
Read/Write	R	R/W	R/W	R/W	R/W	R	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

ADCn_CHx_MUXCTRL, x=0,1,2,3, n=A,B

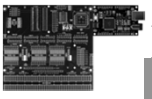


EEL 3744

See doc8331, Section 28.17.5

ADC Result Registers

- For all result registers and with any ADC result resolution, a signed number is represented in 2's complement form, and the MSB represents the sign bit
- The RESL and RESH register pair represents the 16-bit value, ADCRESULT
  - > The low byte of the 16-bit register must be read before the high byte
  - > When the low byte register is read by the CPU, the high byte of the 16-bit register is copied into the temporary register in the same clock cycle as the low byte is read
  - > When the high byte is read, it is then read from the temporary register



EEL 3744 ADC RESH – Channel n Result register High

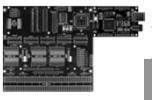
See doc8331, Section 28.17.5

- **12-bit Mode, Left Adjusted**
  - > **RES[11:4]: Channel Result High**
    - – These are the eight MSBs of the 12-bit ADC result
- **12-bit Mode, Right Adjusted**
  - > **RES[11:8]: Channel Result High**
    - – These are the four MSBs of the 12-bit ADC result
- **8-bit Mode**
  - > These bits will be the extension of the sign bit, CHRES7, when the ADC works in signed mode, and set to zero when the ADC works in single-ended mode

ADCn_CHx_RES, x=0,1,2,3, n=A,B

	Bit	7	6	5	4	3	2	1	0
12-bit, left.	+0x05	RES[11:4]							
12-bit, right		–	–	–	–	RES[11:8]			
8-bit		–	–	–	–	–	–	–	–
	Read/Write	R	R	R	R	R	R	R	R
	Initial Value	0	0	0	0	0	0	0	0

0



EEL 3744 ADC RESL – Channel n Result register Low

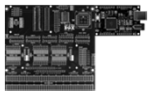
See doc8331, Section 28.17.6

- **12- or 8-bit Mode, Right Adjusted**
  - > **RES[7:0]: Channel Result Low**
    - – These are the eight LSBs of the ADC result
- **12-bit Mode, Left Adjusted**
  - > **RES[3:0]: Channel Result Low**
    - – These are the four LSBs of the 12-bit ADC result

	Bit	7	6	5	4	3	2	1	0
12-/8-bit, right	+0x04	RES[7:0]							
12-bit, left.		RES[3:0]				–	–	–	–
	Read/Write	R	R	R	R	R	R	R	R
	Initial Value	0	0	0	0	0	0	0	0

ADCn_CHx_RES, x=0,1,2,3, n=A,B

41



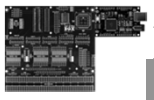
EEL 3744 ADC SCAN – Channel Scan register

See doc8331, Section 28.17.7

- Scan is enabled when COUNT is set differently than 0
- This register is available only for ADC channel 0
- **OFFSET[3:0]: Positive MUX Setting Offset**
  - > The channel scan is enabled when COUNT ≠ 0 and this register contains the offset for the next input source to be converted on ADC channel 0 (CH0)
  - > The actual MUX setting for positive input equals MUXPOS + OFFSET. The value is incremented after each conversion until it reaches the maximum value given by COUNT
  - > When OFFSET = COUNT, OFFSET will be cleared on the next conversion

Bit	7	6	5	4	3	2	1	0	
+0x06	OFFSET[3:0]				COUNT[3:0]				SCAN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

ADCn_CHx_SCAN, x=0,1,2,3, n=A,B



EEL 3744 ADC SCAN – Channel Scan register

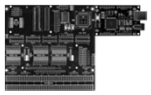
See doc8331, Section 28.17.7

- **COUNT[3:0]: Number of Input Channels Included in Scan**
  - > This register gives the number of input sources included in the channel scan
  - > The number of input sources included is COUNT + 1
  - > The input channels included are the range from MUXPOS + OFFSET to MUXPOS + OFFSET + COUNT

Bit	7	6	5	4	3	2	1	0	
+0x06	OFFSET[3:0]				COUNT[3:0]				SCAN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

University of Florida, EEL 3744 – File **13**
© Drs. Schwartz & Arroyo

ADCn_CHx_SCAN, x=0,1,2,3, n=A,B

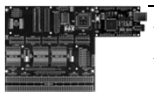


EEL 3744

See doc8331, Section 28.18

ADC Register Summary

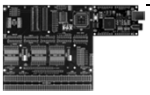
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+0x00	CTRLA	DMASEL[1:0]		CH[3:0]START				FLUSH	ENABLE
+0x01	CTRLB	IMPMODE	CURRLIMIT[1:0]		CONVMODE	FREERUN	RESOLUTION[1:0]		–
+0x02	REFCTRL	–	REFSEL[2:0]			–	–	BANDGAP	TEMPREF
+0x03	EVCTRL	SWEEP[1:0]		EVSEL[2:0]			EVACT[2:0]		
+0x04	PRESCALER	–	–	–	–	–	PRESCALER[2:0]		
+0x05	Reserved	–	–	–	–	–	–	–	–
+0x06	INTFLAGS	–	–	–	–	CH[3:0]IF			
+0x10	CH0RESL	CH0RES[7:0]							
+0x11	CH0RESH	CH0RES[15:8]							
+0x12	CH1RESL	CH1RES[7:0]							
+0x13	CH1RESH	CH1RES[15:8]							
+0x14	CH2RESL	CH2RES[7:0]							
+0x15	CH2RESH	CH2RES[15:8]							
+0x16	CH3RESL	CH3RES[7:0]							
+0x17	CH3RESH	CH3RES[15:8]							
+0x18	CMPL	CMP[7:0]							
+0x19	CMPH	CMP[15:8]							



EEL 3744

ADC Register Requirements For Our Lab

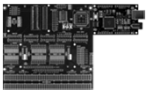
- Enable ADC: **ADCA_CTRLA**
- Set reference: **ADCA_REFCTRL**
  > You need to use the value that will make VREF=External AREF pin or PORTB
- Set sample time: **ADCA_PRESCALER**
  > Al Gore suggests using DIV512
- Set mode (unsigned or signed; single scan or free running) **ADCA_CTRLB**
  > I suggest FREERUN and 8-bit right-adjusted
- Set ADC pin for input: **PORTA_DIR**
- Start the scan: **ADCA_CH0_CTRL**
- Wait for result (or use interrupts): **ADCA_CH0_INTFLAGS**
- Get Result: **ADCA_CH0_RES**
  > This might be two bytes to deal with, depending on the number of bits and right- or left-adjusted



EEL 3744

A/D Definitions

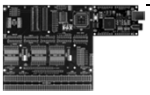
- **Analog**: Continuous in time and voltage
- **Digital**: Discrete in time (sampling) and voltage (a fixed set of possible values, e.g., if 3 bits, then 2^3=8 possible values)
- **Span**: Range of possible analog voltages
  - > Span = $V_H - V_L$
  - > If V_H = 5 V and V_L = 0 V, **Span = 5V**
- **Resolution** (Δ): Smallest change in an input that will produce a change in the output
  - > Δ = Span / 2^n , where n is the number of bits
    - – If 2.37 V to 2.38 V is the smallest change allowed, Δ = 0.01 V
  - > If V_H=5V & V_L=0, and n = 8, then **Δ = 5V / 2^8 = 19.5 mV**
    - – If n = 16, **Δ = 5 V / 2^{16} = 76.2 μV**



EEL 3744

A/D Definitions

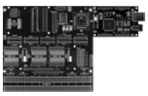
- **Dynamic Range**:
  - > D.R. = Largest Voltage / Smallest Voltage
  - > D.R. = V_{max} / $V_{min\text{-}measurable}$, often measured in dB
    - – $V_{min\text{-}measurable}$ = Δ
  - > For noise in a system, replace Δ with V_{noise}
  - > $D.R._{dB}$ = 20 log(V_{max} / $V_{min\text{-}meas}$) = 20 log(V_{max} / Δ)
  - > $D.R._{dB}$ = 20 log(V_{max} / [Span / 2^n])
  - > If V_{min}=0, then $D.R._{dB}$ = 20 log(V_{max} / [V_{max} / 2^n]) ➔ $D.R._{dB}$ = 20 log(2^n) = n × 20 log (2) ≈ 6 × n dB
  - > Example: Given 8-bit A/D, range of 0V to 5V
    - – Span = $V_H - V_L$ = 5 – 0 = **5V**
    - – Δ = Span / 2^n = 5V / 2^8 = **19.5 mV**
    - – $D.R._{dB}$ = 20 log(V_{max} / Δ) = 20 log(5V / 19.5mV) = **48.2 dB**
    - – $D.R._{dB}$ ≈ 6n dB = 6*8 dB = **48 dB**



EEL 3744

A/D Definitions

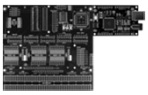
- **Accuracy**: Closeness of a measurement to its actual value
  - > Example for resolution, Δ = 19.5 mV
    - – If measured 37 mV, then 100% * 19.5/37 = 52.7% (pretty bad!)
    - – If measured 370 mV, then 100% * 19.5/370 = 5.27%
    - – If measured 3.7 V, then 100% * 0.0195/3.7 = 0.527%
- Nyquist-Shannon Theorem: Sampling frequency must be at least twice the highest frequency (in order to properly reconstruct the original signal)
  - > $f_{sample} \geq 2 \times f_{max}$ ➔ $T_{sample} \leq 1 / (2\ f_{max})$
    - – T_{sample} is the maximum A/D conversion time necessary to accurately reproduce the original signal



EEL 3744

Sampling Rates

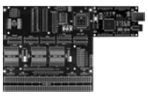
- Telephone (narrowband): **8 kHz**
  - > Wideband telephone, VoIP, VVoIP: **16 kHz**
- MPEG Audio: **11.025 kHz**
- Audio CDs sample at **44.1 kHz** (and uses 16-bits)
- Profession audio sampling rate using tape recorders, video servers, etc: **48 kHz**
- First commercial digital audio recorders (1970s): **50 kHz**
- Pro recording equipment for making CDs: **88.2 kHz**
- DVD-audio, Blue-ray disk audio, HD DVD audio: **96kHz**
- Recording equipment for DVD-audio, Blue-ray disk audio, HD DVD audio: **192 kHz**
- **Noise kills dynamic range**



EEL 3744

Dynamic Range Examples

- Dynamic Ranges for various systems
  - >8-track tapes: 50 dB
  - >Dolby B: 62 dB
  - >CDs (16-bit): 96 dB (theoretical)
  - >Digital Audio (16-bit): 96 dB (theoretical)
    - Observed 16-bit digital audio: 90 dB
  - >Digital Audio (20-bit): 120 dB (theoretical)
  - >Digital Audio (24-bit): 144 dB (theoretical)



EEL 3744

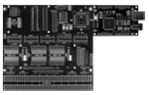
68HC12 ATDCTL2: ATD Control Register 2 (A/D Power-Up)

	7	6	5	4	3	2	1	0	ATDCTL2
$0062	**ADPU**	**AFFC**	AWAI	0	0	0	ASCIE	ASCIF	
RESET	0	0	0	0	0	0	0	0	

A/D Power-Up
0 = Off
1 = On

A/D Fast Flag Clear
0 = A/D flag clears normally (read status register than result register)
1 = A/D flags fast clear (access to result register will clear associate flag)

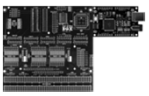
- It takes **100µs** for the charge pump to stabilize, so turn A/D power on at least **200** E-clocks (for E=2 MHz, 0.5µs) **before** use



EEL 3744 **68HC12** ATDCTL2: ATD Control Register 2 (A/D Power-Up)

	7	6	5	4	3	2	1	0 ATDCTL2
$0062	ADPU	AFFC	**AWAI**	0	0	0	**ASCIE**	**ASCIF**
RESET	0	0	0	0	0	0	0	0

- AWAI: A/D stop in wait mode
- ASCIE: A/D sequence complete interrupt enable
  >0 = disables A/D interrupt; 1 = enables A/D interrupt
- ASCIF: A/D sequence complete interrupt flag
  >0 = no A/D interrupt; 1 = A/D interrupt occurred

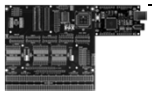


EEL 3744 **68HC12** ATDCTL4: A/D Control Register 4 (Sample and Conversion Times)

	7	6	5	4	3	2	1	0 ATDCTL4
$0064	0	SMP1	SMP0	PRS4	PRS3	PRS2	PRS1	PRS0
RESET	0	0	0	0	0	0	0	1

S&HE: Fig 12.2 & Fig 12.3 & Tab 12.1 & Tab 12.2

- PRS4:0 are the A/D prescaler bits for the conversion times
  > Set these bits so the A/D clock frequency is between 2 MHz and 500 KHz
    – For our board with E = 2 MHz, PRS=0 (=> 1 MHz) or 1 (=> 500 KHz)
- SMP1:0 are the A/D sample time bits and determine the time a signal is sampled (see S&HE: Fig 12.2 & Tab 12.2)
  > This will determine the total conversion time in E-clocks
    – Minimum is 18 E-clocks; maximum is 32 E-clocks (see S&HE: Tab 12.2)
- Normally, leave these as all at zero (or the PRS0=1)



EEL 3744 **68HC12** ATDCTL5: A/D Control Register 5 (Scanning and Channels)

	7	6	5	4	3	2	1	0 ATDCTL5
$0065	0	S8CM	SCAN	MULT	CD	CC	CB	CA
RESET	0	0	0	0	0	0	0	0

Channel Select

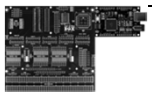
Continuous **Scan** Control
0=Scan once
1=Scan continuously
(both through 4 channels)

Channel Control
0 = Single Channel
1 = Multiple Channel

Tech: Tab 17.4

S&HE: Tab 12.3

S8CM (Select 8-channel mode)
0 = 4 conversions
1 = 8 conversions

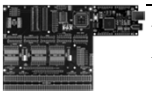


EEL 3744 **68HC12** A/D Channel Assignment

- If MULT=0 (single channel selected)
  - >CD should be 0
  - >CC:CA determine the single selected channel
    - – Example CC:CA=101 => A/D channel 5 is selected
- If MULT=1 (multiple channels selected)
  - >4-channel conversion
    - – CB & CA have no effect
    - – CC determines the group of four channels used (CD is ignored)
  - >8-channel conversion
    - – CC, CB & CA have no effect
    - – CD should be 0
- **See Tech: Tab 17.4**

Tech:Tab 17.4

S&HE:Tab 12.3

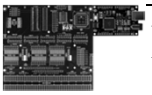


EEL 3744 **68HC12** ATDSTAT: A/D Status Registers (High Byte)

ATDSTATH

	7	6	5	4	3	2	1	0
$0066	SCF	0	0	0	0	CC2	CC1	CC0
RESET	0	0	0	0	0	0	0	0

- SCF: Sequence complete flag
  - >Set when a conversion sequence (4 or 8 samples) have been taken (when SCAN=0) or when first sequence has been taken (when SCAN=1)
- CC2-CC0: Conversion counter for the current sequence
  - >CC2:0 give the binary code of the register that will be written next

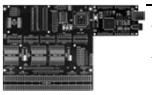


EEL 3744 **68HC12** ATDSTAT: A/D Status Registers (Low Byte)

ATDSTATL

	7	6	5	4	3	2	1	0
$0067	CCF7	CCF6	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0
RESET	0	0	0	0	0	0	0	0

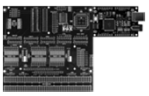
- CCF7-CCF0: Conversion complete flags
  - >Each flag is associated with an A/D result register
  - >Each bit is set at the end of the conversion
  - >Clear a bit by one of the below techniques
    - – If AFFC in ADCTL2 is clear (fast clear = 0, default)
      - ☞Clear by reading ATDSTAT then reading the corresponding A/D result register
    - – If AFFC in ADCTL2 (fast clear = 1)
      - ☞Clear by reading the corresponding A/D result register (no read of ADSTAT is necessary)



EEL 3744 **68HC12** ADR0H-ADR7H: A/D Result Registers

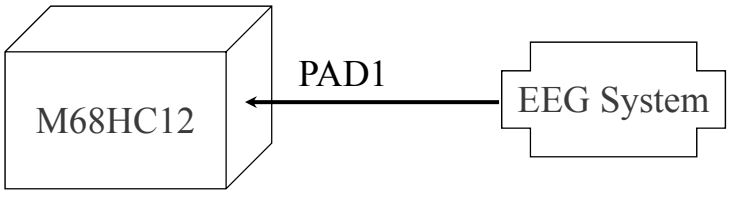
- High A/D result registers are used for 8-bit A/D
- These registers are read only

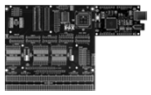
	7	6	5	4	3	2	1	0	
$0070	Bit7	-	-	-	-	-	-	Bit0	ADR0H
$0072	Bit7	-	-	-	-	-	-	Bit0	ADR1H
○ ○ ○									○ ○ ○
$007C	Bit7	-	-	-	-	-	-	Bit0	ADR6H
$007E	Bit7	-	-	-	-	-	-	Bit0	ADR7H



EEL 3744 A/D EEG Example with 68HC12

- Problem Statement
  - Collect 100 samples of an EEG signal sampled at 125Hz
  - Place the 100 samples starting at location EEG
  - The EEG signal is amplified and is presented as a 0-5V level analog signal to the A/D



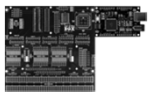


EEL 3744

A/D EEG Example

- f_S=125Hz, T_S = 1/125 = 0.008s = 8ms, i.e., collect a sample every 8ms
- Let us use the RTI (real-time interrupt) system to generate the timing
  - > If RTICTL2:0 (RTR2:RTR0) are 010 then the system interrupts every 8.192ms for E=2 MHz (i.e., f = 1/T = 122Hz ≈ 125 Hz)
  - > Q: What would you do if we need a sample every 370μs?
- We'll use PAD1 and therefore set ATDCTL53:0 to 0001
- MULT=0 (single channel) ; SCAN=0 (scan once); S8CM=0 (4-channels)
- We will assume the data is noisy, so we'll take 4 quick samples and record the average of the four values
- ISR for RTI is (without D-Bug12) located at $FFF0

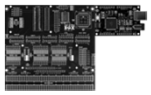
AD_EEG.asm



EEL 3744

Analog/Digital Conversion

- A/D Conversion Method in the M68HC11
  - >Charge Distribution A/D (see RM Chapter 12)
    - – The 68HC12's A/D is similar



EEL 3744

Basic Charge-Redistribution A/D

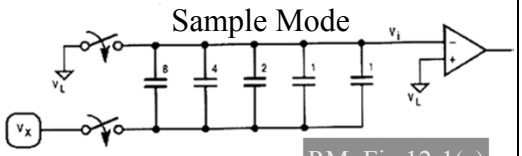
a) Sample Mode

the Total Charge: $\mathbf{Q_S = C\ V}$

$Q_S = 16\ (V_X - V_L)$

$\therefore Q_S = 16\ V_X$ (with $V_L = 0$)

Sample Mode

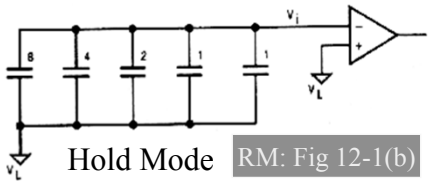


RM: Fig 12-1(a)

b) Hold Mode (see figure, next column)

$Q_H = (V_L - V_i)\ 16$

$\therefore Q_H = -16\ V_i$ (with $V_L = 0$)



Hold Mode RM: Fig 12-1(b)

Since charge is conserved,

$Q_S = Q_H$

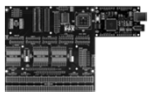
$16\ V_X = -16 V_i$

$V_X = -V_i$

$\therefore V_i = -V_X$

c) Approximation Mode

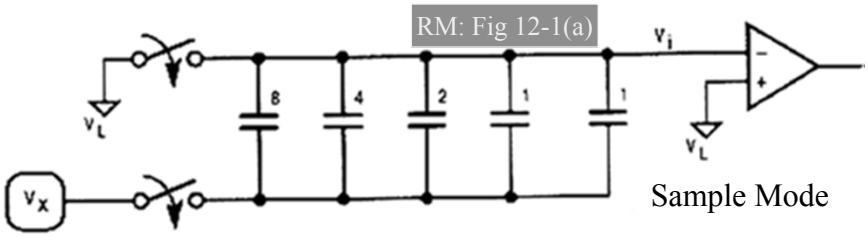
RM: Fig 12-1(c)



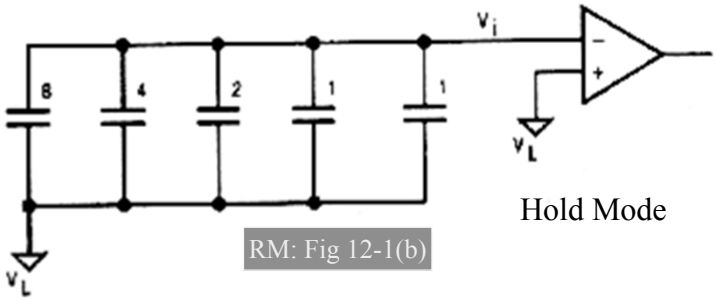
EEL 3744

Sample Mode & Hold Mode

RM: Fig 12-1(a)

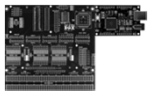


Sample Mode

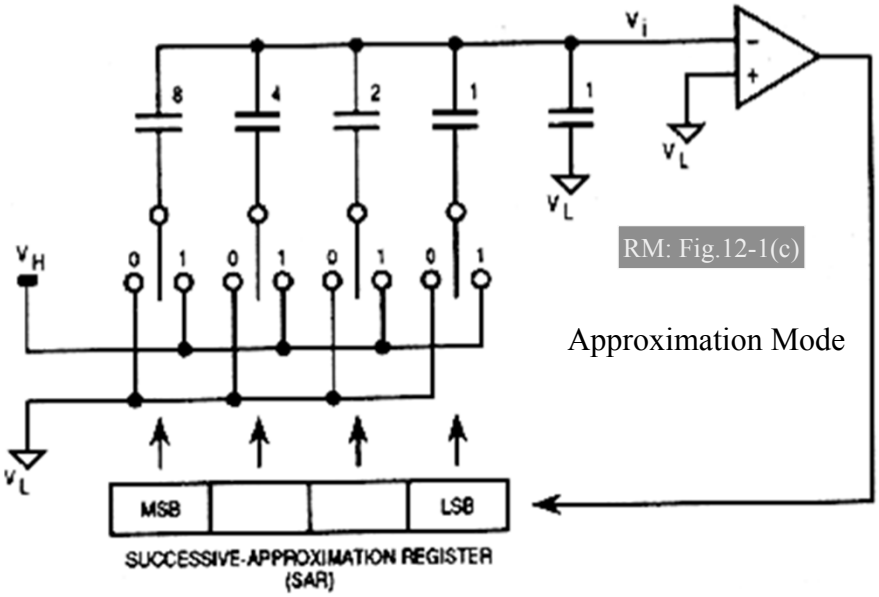


Hold Mode

RM: Fig 12-1(b)

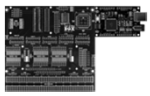


EEL 3744
Approximation Mode



RM: Fig.12-1(c)

Approximation Mode



EEL 3744
A Conversion Sequence Example (4 bit)

Let $V_X = 21/32\ V_H$

During the sample time,

$Q_S = 16\ V_X = (16)\ 21/32\ V_H$

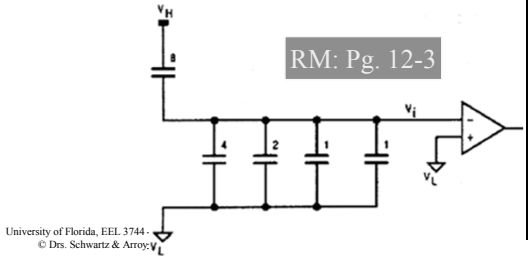
$\therefore Q_S = 21/2\ V_H$

During hold,

$\therefore\ V_i = -V_X = -21/32\ V_H$

• 8-unit capacitor: $V_L \rightarrow V_H$

$Q = 8\ (V_H - V_i) + 8\ (V_L - V_i)$



RM: Pg. 12-3

with $V_L = 0 \Rightarrow$

$Q = 8V_H - 16\ V_i$

By charge conservation, this charge is set equal to the original charge ($Q_S = Q$):

$21/2\ V_H = 8V_H - 16\ V_i$

Solving for V_i,

$16\ V_i = 8V_H - 21/32\ V_H$

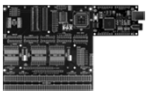
$\therefore V_i = -5/32\ V_H$

(when $V_i<0$, comparator output = 1; when $V_i>0$, comparator output = 0)

$\therefore$ the output of comparator = 1

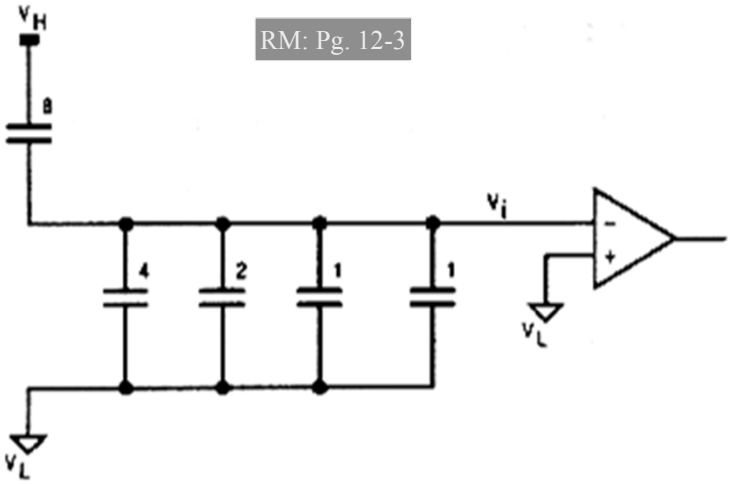
$\Rightarrow \mathbf{1}???_2$ (and leave 8-unit cap at V_H)

65

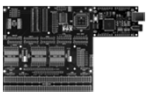


EEL 3744

Eight-unit capacitor switched from low to high



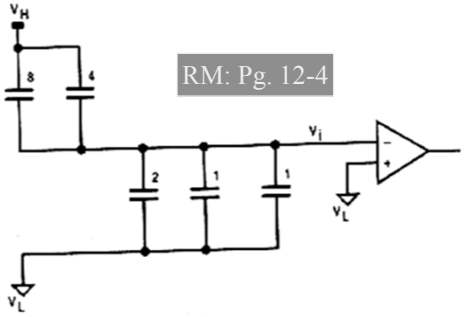
RM: Pg. 12-3



EEL 3744

A Conversion Sequence Example (4 bit)

> 4-unit capacitor: $V_L \rightarrow V_H$

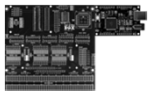


RM: Pg. 12-4

$Q = 12\,(V_H - V_i) - 4\,(V_i - V_L)$
with $V_L = 0 \Rightarrow$
$Q = 12V_H - 16\,V_i$

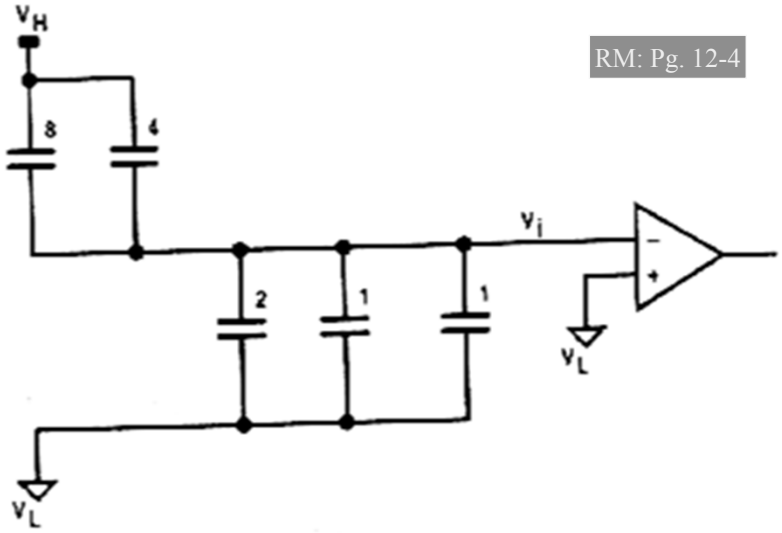
By charge conservation,
$21/2\ V_H = 12V_H - 16\ V_i$
Solving for V_i,
$\therefore V_i = 3/32\ V_H$
(when $V_i<0$, comparator output = 1;
when $V_i>0$, comparator output = 0)
$\therefore$ the output of comparator = 0
$\Rightarrow$ **10**??$_2$ (and reconnect 4-unit cap to V_L)

$\therefore$ Digital result of this example ($21/32\ V_H$) conversion $\Rightarrow$ **10**??$_2$

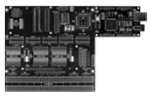


EEL 3744

Try Four-unit Capacitor



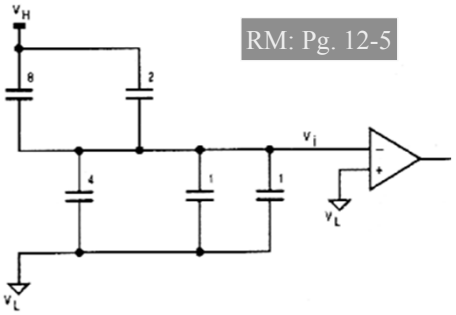
RM: Pg. 12-4



EEL 3744

A Conversion Sequence Example (4 bit)

> 2-unit capacitor: $V_L \rightarrow V_H$



RM: Pg. 12-5

$Q = 10\,(V_H - V_i) - 6\,(V_i - V_L)$

with $V_L = 0 \Rightarrow$

$Q = 10V_H - 16\,V_i$

By charge conservation,

$21/2\ V_H = 10V_H - 16\ V_i$

Solving for V_i,

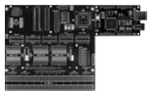
$\therefore V_i = -1/32\ V_H$

(when V_i<0, comparator output = 1;
when V_i>0, comparator output = 0)

$\therefore$ the output of comparator = 1

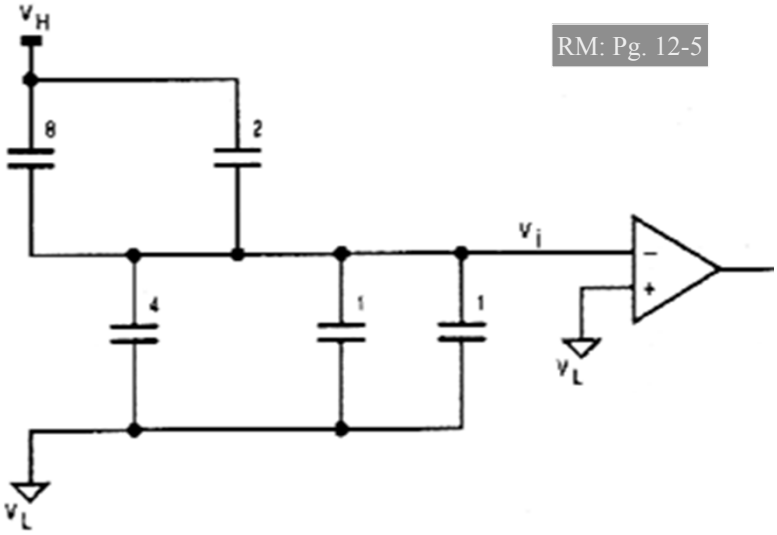
$\Rightarrow 101?_2$ (and leave 2-unit cap at V_H)

$\therefore$ Digital result of this example (21/32 V_H) conversion $\Rightarrow$ **101**$?_2$

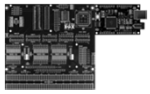


EEL 3744

Try Two-unit capacitor



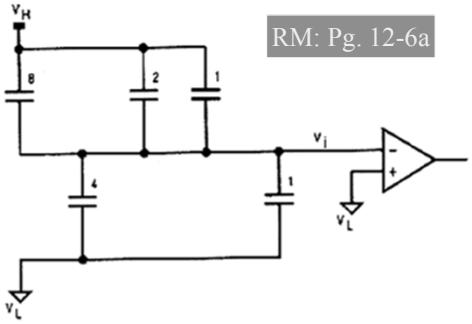
RM: Pg. 12-5



EEL 3744

A Conversion Sequence Example (4 bit)

> 1-unit capacitor: $V_L \rightarrow V_H$

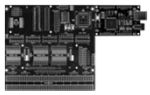


RM: Pg. 12-6a

$Q = 11\ (V_H - V_i) - 5\ (V_i - V_L)$
with $V_L = 0 \Rightarrow$
$Q = 11V_H - 16\ V_i$

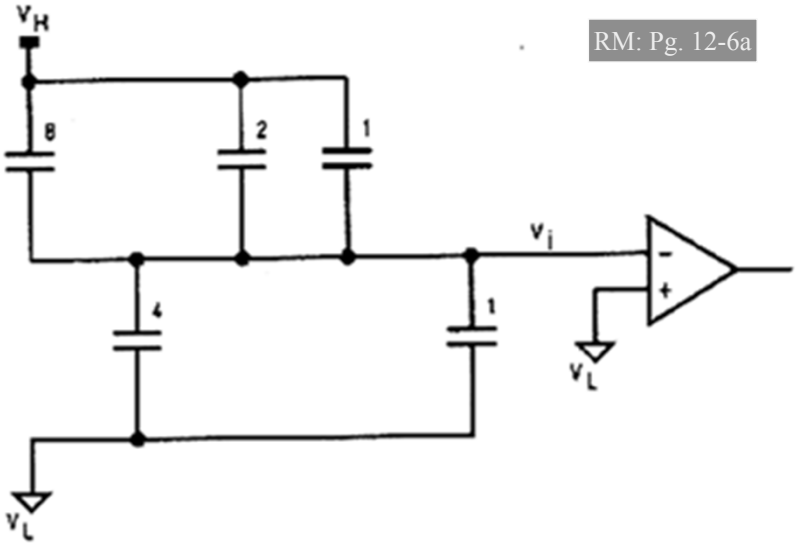
By charge conservation,
$21/2\ V_H = 11V_H - 16\ V_i$
Solving for V_i,
$V_i = 1/32\ V_H$
(when $V_i<0$, comparator output = 1;
when $V_i>0$, comparator output = 0)
$\therefore$ the output of comparator = 0
$\Rightarrow 101\mathbf{0}_2$ (done!)

$\therefore$ Digital result of this example ($21/32\ V_H$) conversion $\Rightarrow$ $\mathbf{1010}_2$ ($10/16\ V_H$)

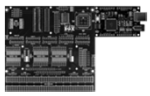


EEL 3744

Try One-unit capacitor



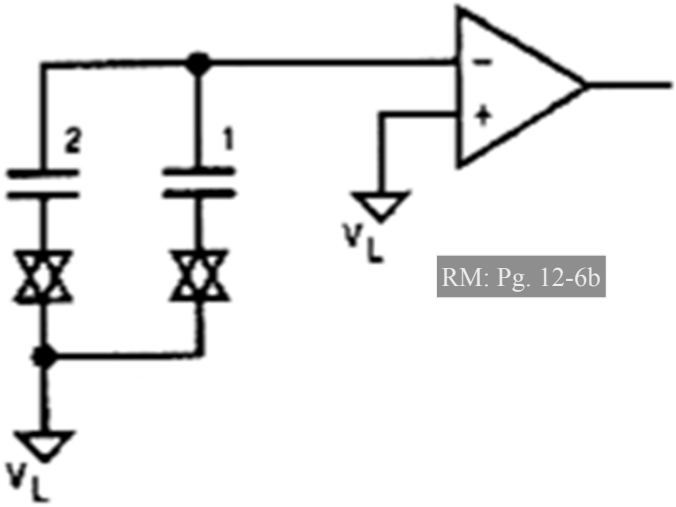
RM: Pg. 12-6a



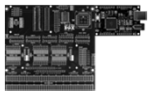
EEL 3744

A Simple 2-bit A/D

A/D if the 2nd 1-unit capacitor is omitted



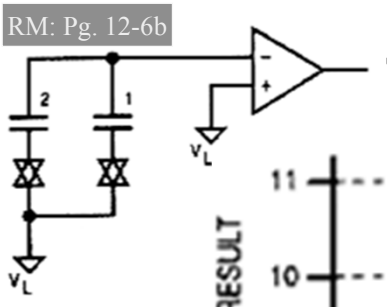
RM: Pg. 12-6b



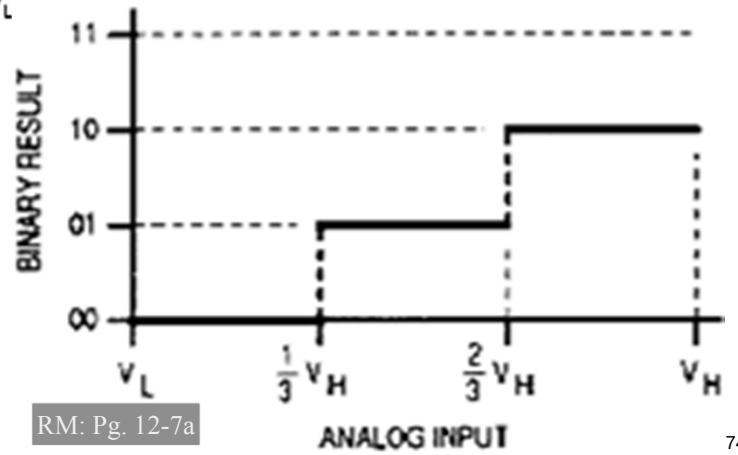
EEL 3744

A Simple 2-bit A/D

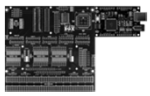
RM: Pg. 12-6b



Transfer characteristic if the 2nd 1-unit capacitor is omitted



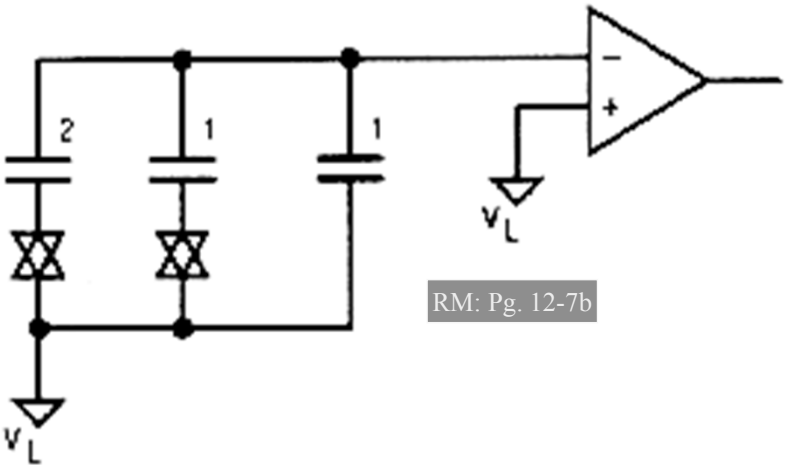
RM: Pg. 12-7a



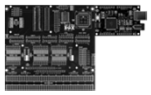
EEL 3744

A Simple 2-bit A/D

A/D if the 2nd 1-unit capacitor is added

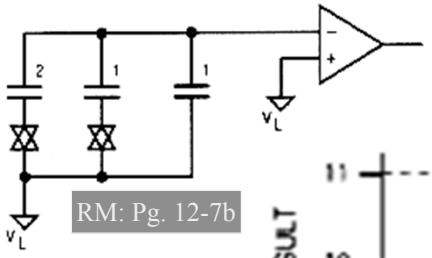


RM: Pg. 12-7b



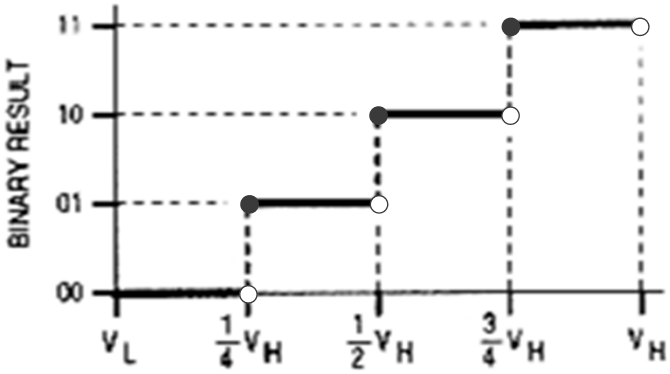
EEL 3744

A Simple 2-bit A/D

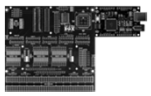


RM: Pg. 12-7b

Transfer characteristic if the 2nd 1-unit capacitor is added

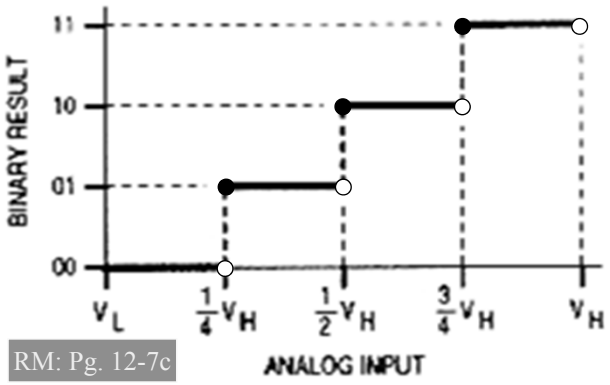


RM: Pg. 12-7c



EEL 3744

A Simple 2-bit A/D



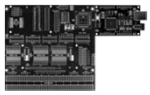
RM: Pg. 12-7c

Example:

(ex) $V_X = 1/4\ V_H \Rightarrow 01_2\ (1/4\ V_H)$

$V_X = 1/8\ V_H \Rightarrow 00_2\ (0\ V_H)$; error by $1/8\ V_H$ or 1/2 LSB

$\therefore$ This 2-bit A/D has a Quantization Error of -0/+1 LSB



EEL 3744
Charge-Redistribution A/D with ±1/2 LSB Quantization Error

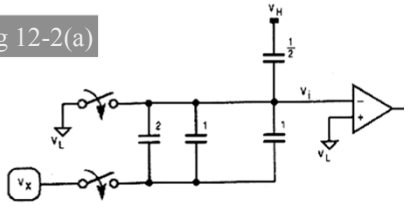
a) Sample Mode

$Q_S = 4(V_X - V_L) + 1/2\,(V_H - V_L)$

with $V_L = 0 \Rightarrow$

$\therefore Q_S = 4V_X + 1/2\,V_H$

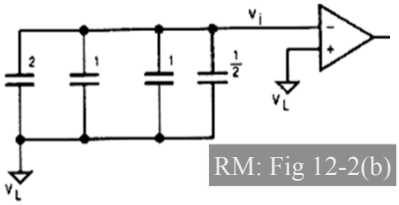
RM: Fig 12-2(a)



b) Hold Mode

$Q_H = 9/2\,(V_L - V_i)$

$\therefore Q_H = -9/2\,V_i$ (with $V_L = 0$)



RM: Fig 12-2(b)

Since charge is conserved,

$Q_S = Q_H$

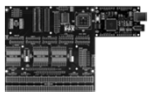
$4V_X + 1/2\,V_H = -9/2\,V_i$

$8V_X + V_H = -9V_i$

$\therefore 9/8V_i = -V_X - 1/8V_H$

c) Approximation Mode

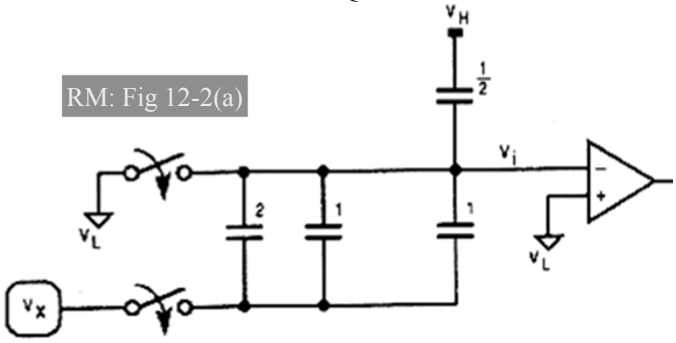
RM: Fig 12-2(c)



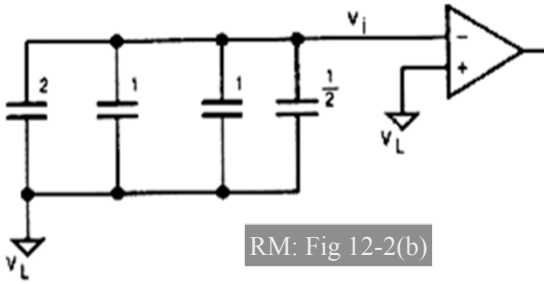
EEL 3744
Charge-Redistribution A/D with ±1/2 LSB Quantization Error

a) Sample Mode

RM: Fig 12-2(a)



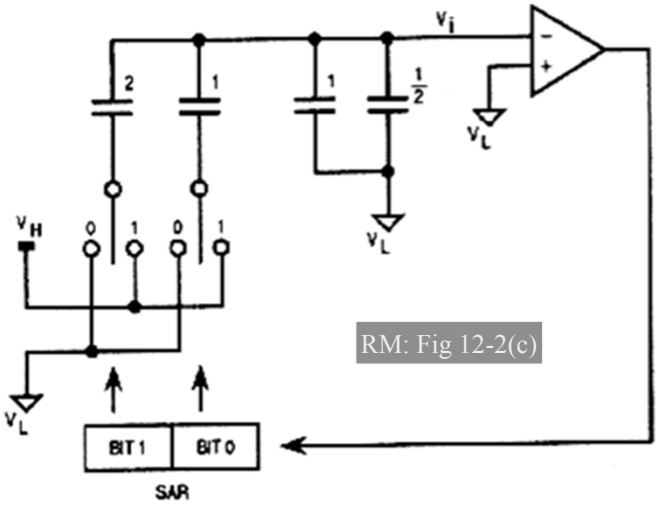
b) Hold Mode



RM: Fig 12-2(b)

EEL 3744 Charge-Redistribution A/D with ±1/2 LSB Quantization Error

c) Approximation Mode



RM: Fig 12-2(c)

EEL 3744 Charge-Redistribution A/D with ±1/2 LSB Quantization Error

> The Equivalent circuit for a digital result of 01_2:

$Q = (V_H - V_i) - 7/2\ (V_i - V_L)$

with $V_L = 0 \Rightarrow$

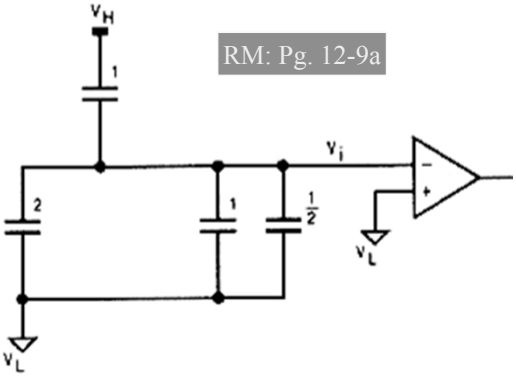
$Q = V_H - 9/2\ V_i$

By charge conservation,

$4\ V_X + 1/2\ V_H = V_H - 9/2\ V_i$

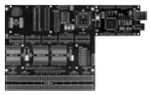
Solving for V_i,

$\therefore 9/8 V_i = 1/8 V_H - V_X$



RM: Pg. 12-9a

$\therefore$ the output of comparator = **1** ($\Rightarrow 0\mathbf{1}_2$) if $V_X > 1/8\ V_H$



EEL 3744
Charge-Redistribution A/D with ±1/2 LSB Quantization Error

> The Equivalent circuit for a digital result of 10_2:

$Q = 2\,(V_H - V_i) - 5/2\,(V_i - V_L)$

with $V_L = 0 \Rightarrow$

$Q = 2\,V_H - 9/2\,V_i$

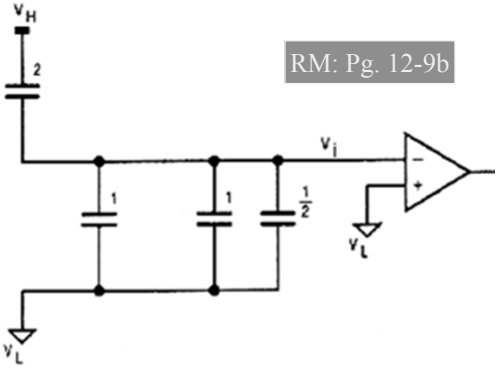
By charge conservation,

$4\,V_X + 1/2\,V_H = 2\,V_H - 9/2\,V_i$

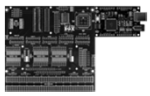
Solving for V_i,

$\therefore 9/8 V_i = 3/8 V_H - V_X$

$\therefore$ the output of comparator = **1** ($\Rightarrow$ **1**0_2) if $V_X > 3/8\,V_H$



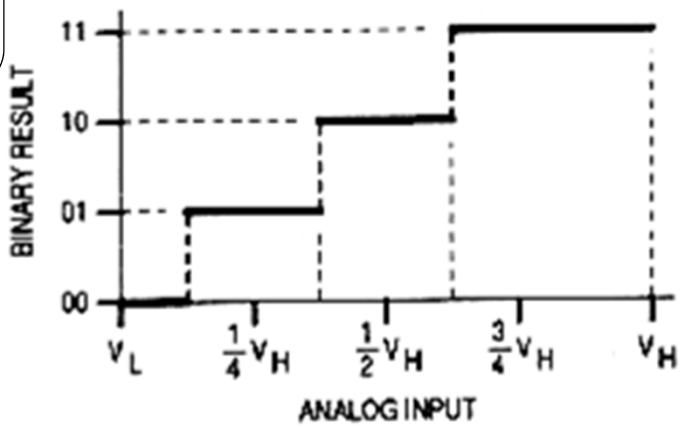
RM: Pg. 12-9b



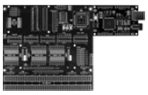
EEL 3744
Charge-Redistribution A/D with ±1/2 LSB Quantization Error

NOTE:
There is no 2-bit digital code for 4/4 V_H (full scale)

the complete transfer characteristic of the circuit

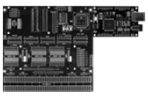


RM: Pg. 12-10



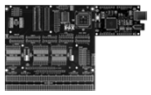
EEL 3744
Quantization in 2 bits

- Suppose your A/D Converter yields 2 bits
  - >What should the answers be?
  - >There are 4 bit patterns possible, mainly {00,01,10,11}:
    - **00** is the bit pattern for VRL {$\mathbf{0 \le V_{\%} < 0.25}$}
    - **01** for 25% or 1/4 of (VRH-VRL) {$\mathbf{0.25 \le V_{\%} < 0.50}$}
    - **10** for 50% or 1/2 of (VRH-VRL) {$\mathbf{0.50 \le V_{\%} < 0.75}$}
    - **11** for 75% or 3/4 of (VRH-VRL) {$\mathbf{0.75 \le V_{\%} < 1.00}$}
  - >Let V_{RH}=5V and V_{RL}=0V and if our unknown voltage is:
    - V_x=2.00V, then answer will be 01 for $V_{\%}$= 2.00/5 or 40% of V_{RH}
    - V_x=1.25V, then answer will be 01 for $V_{\%}$= 1.25/5 or 25% of V_{RH}
    - Vx=4.00V, then answer will be 11 for $V_{\%}$= 4.00/5 or 80% of V_{RH}



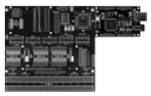
EEL 3744
Quantization in 3 bits

- Suppose your A/D Converter yields 3 bits
  - >There are 4 bit patterns possible, mainly {00,01,10,11}:
    - **000** is the bit pattern for VRL {$\mathbf{0 \le V\% < 0.125}$}
    - **001** for 12.5% or 1/8 of (V_{RH}-V_{RL}) {$\mathbf{0.125 \le V_{\%} < 0.250}$}
    - **110** for 75% or 6/8 of (V_{RH}-V_{RL}) {$\mathbf{0.75 \le V_{\%} < 0.875}$}
    - **111** for 87.5% or 7/8 of (V_{RH}-V_{RL}) {$\mathbf{0.875 \le V_{\%} < 1.00}$}
  - >Let V_{RH}=5V and V_{RL}=0V and if our unknown voltage is:
    - V_x =2.00 V, the answer will be 011 for $V_{\%}$=2.00/5 or 40% of V_{RH}
    - V_x=1.25 V, the answer will be 010 for $V_{\%}$=1.25/5 or 25% of V_{RH}
    - V_x=4.00 V, the answer will be 110 for $V_{\%}$=4.00/5 or 80% of V_{RH}



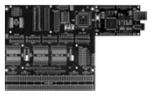
EEL 3744
Quantization in 4 bits

- Suppose your A/D Converter yields 4 bits
  - >There are 16 bit patterns possible, mainly {0000,0001,...,1110,1111}:
    - 0000 is the bit pattern for VRL {**0≤ V%<0.0625**}
    - 0001 for 6.25% or 1/16 of (V_{RH}-V_{RL}) {**0.0625≤ V%<0.125**}
    - 1110 for 87.5% or 14/16 of (V_{RH}-V_{RL}) {**0.875≤ V%<0.9375**}
    - 1111 for 93.75% or 15/16 of (V_{RH}-V_{RL}) {**0.9375≤ V%<1.00**}
  - >Let V_{RH}=5V and V_{RL}=0V and if our unknown voltage is:
    - V_x=2.00V, the answer will be 0110 for $V_{\%}$=2.00/5 or 40% of V_{RH}
    - V_x=1.25V, the answer will be 0100 for $V_{\%}$=1.25/5 or 25% of V_{RH}
    - V_x=4.00V, the answer will be 1100 for $V_{\%}$=4.00/5 or 80% of V_{RH}



EEL 3744
Quantization in 8 bits

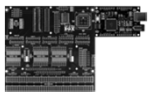
- Now, if your A/D Converter yields 8 bits
  - >There are 256 bit patterns possible, {00000000,00000001,...,11111110,11111111}
  - >Thus, 00000000 is the bit pattern for V_{RL} {**0 ≤ $V_{\%}$ < 0.00390625**}
  - >00000001 for 0.390625% or 1/256 of (V_{RH}-V_{RL})
  - >11111111 for 99.609375% or 255/256 of (V_{RH}-V_{RL})
  - >Etc.



EEL 3744

Checking Your A/D

- If your A/D Converter yields 8 bits
  - > And there are 256 bit patterns possible, mainly {00000000,00000001,...,11111110,11111111} Thus,
    - 00000000 is the bit pattern for V_{RL} {**$0 \le V_{\%} < 0.00390625$**}
    - 11111111 for 99.609375% or 255/256 of (V_{RH}-V_{RL})
    - 01001100 for 29.6875% or 76/256 of (V_{RH}-V_{RL})
  - > Then if you connect a "C" battery (with normal voltage of 1.5V) to, an ADC pin with V_{RH}=5V and V_{RL}=0V, then the A/D should yield \$4C = 76_{10} = %01001100
  - > If you get \$46 you are probably satisfied ...
  - > If you get \$35, your battery is probably dead …
  - > But if you get larger than say \$50, for example, \$DE, then something is VERY wrong!!! How can a "C" battery >> 1.5V?



EEL 3744

The End!