Mixed Signal System

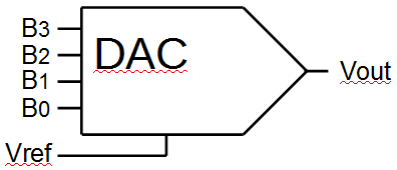
Mixed Signal Integrated Circuit
any integrated circuit that has both analog circuits and digital circuits on a single semiconductor die.

Embedded System
a computer system—computer processor + memory + I/O peripheral devices—that has a dedicated function within a stand-alone product or as a sub-function within a larger system.

D/A Conversion Topics

DAC quantization: Conversion of binary-value input signals to a discrete-value output voltage
Resolution: Defines the # of discrete quantized levels resolved by the converter

DAC resolution
Bit resolution, N, equals the number of binary-weighted DAC inputs
Voltage Resolution, $V_{LSB} = V_{REF}/2^N$ smallest resolvable V_{OUT} value
divides the Vref range into 2^N discrete values
Full-scale V_{OUT} = $V_{LSB} * (2^N - 1) = V_{REF} - V_{LSB}$
V_{OUT} = $V_{LSB} * (\text{Input code})_d = (V_{REF}/2^N) * (\text{Input code})_d$



For a 4-bit D/A converter with V_{REF} = 16V

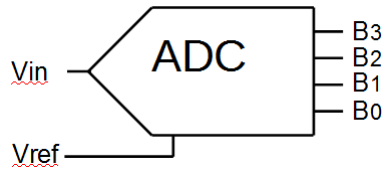
What is the V_{LSB}	V_{LSB} = 16V/16 = 1V
What is the V_{FS}	V_{FS} = 1V*15 = 16V – 1V = 15V
What is V_{OUT} @ B = 0x2	V_{OUT} = 1V * $2_{DECIMAL}$ = 2V
What is V_{OUT} @ B = 0x8	V_{OUT} = 1V * $8_{DECIMAL}$ = 8V
What is V_{OUT} @ B = 0xC	V_{OUT} = 1V * $12_{DECIMAL}$ = 12V

Binary-Weighted DAC Inputs

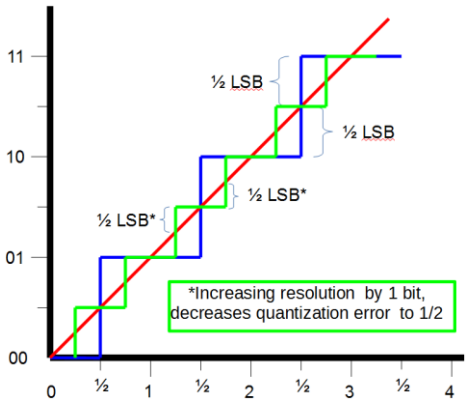
$B_3 = 1$	=> $V_{REF}/2$
$B_2 = 1$	=> $V_{REF}/4$
$B_1 = 1$	=> $V_{REF}/8$
$B_0 = 1$	=> $V_{REF}/16$

A/D Conversion Topics

ADC quantization: Conversion of continuous-value input signals to discrete-value binary out
Resolution: Defines the # of discrete quantized levels resolved by the converter.
ADC resolution
Bit resolution, N, equals the number of binary-weighted ADC outputs
Voltage Resolution, $V_{LSB} = V_{REF}/2^N$ smallest resolvable V_{IN} value
divides the Vref range into 2^N discrete values
Full-scale V_{IN} = $V_{LSB} * (2^N - 1) = V_{REF} - V_{LSB}$
B_{OUT}** = $(V_{IN}/V_{LSB})_b = (2^N * V_{IN}/V_{REF})_b$ ranges from 0 to 2^N-1 binary
** round to nearest integer value (quantization error)



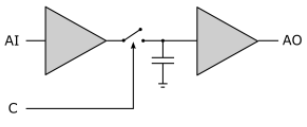
A/D quantization error:
Difference between the input analog signal & the closest equivalent discrete digital output value at each sampling instant.
Quantization error introduces noise - quantization noise - to the sample signal.
Increasing the resolution decreases the quantization noise.



For a 4-bit A/D converter with V_{REF} = 5V

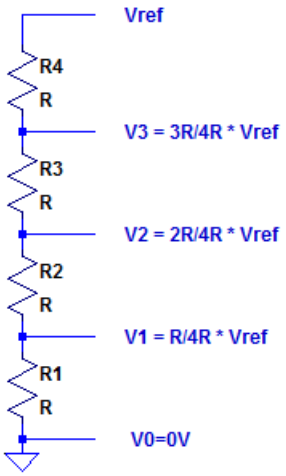
What is the V_{LSB}	V_{LSB} = 5V/16 = 0.3125V
What is the V_{FS}	V_{FS} = 0.3125V*15 = 5V – 0.3125V = 4.6875V
What is B_{OUT} @ V_{IN} = 0.625V	B_{OUT} = 0.625V/0.3125V = $2_{DECIMAL}$ = 0x2
What is B_{OUT} @ V_{IN} = 2.500V	B_{OUT} = 2.500V/0.3125V = $8_{DECIMAL}$ = 0x8
What is B_{OUT} @ V_{IN} = 3.750V	B_{OUT} = 3.750V/0.3125V = $12_{DECIMAL}$ = 0xC

Sample-Hold
Captures and stores an instantaneous sample of an analog signal for a specified time.
Uses a switched capacitor circuit.
Acquisition time min time required to acquire the sample to the specified accuracy
Voltage Droop rate of voltage decay per time, ie µV/µS.



Resistor Ladder Nets

String Ladder:

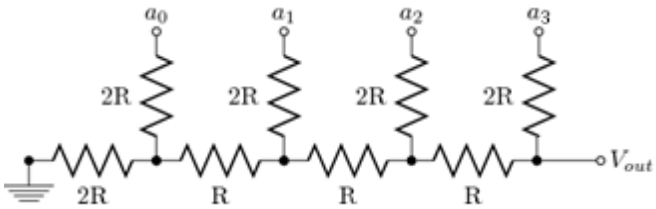


voltage divider used for A/D Flash conversion
requires 2^N resistors to achieve an N-bit conversion
partitions V_{REF} into 2^N discrete values
With equal value resistors R
$V_{LSB} = V_{REF}/2^N$
V_{OUT} (full-scale) = $V_{LSB} * (2^N-1) = V_{REF} - V_{LSB}$

simple network, implements fast A/D conversion
but ….. requires 2^N resistors to achieve an N-bit conversion

2-Bit example, 4 Resistors, V_{LSB} is $V_{REF}/4$

R-2R Ladder:



4-Bit R-2R Ladder Network

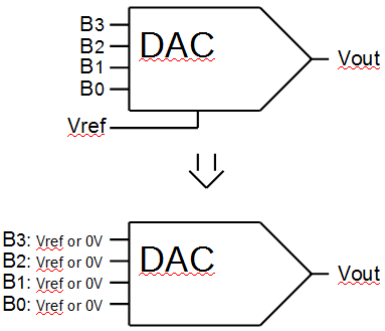
voltage divider used for D/A conversion
requires 2*N resistors to achieve an N-bit conversion
N equals the # of R-2R inputs, a_0-a_{N-1}.
partitions V_{REF} into 2^N discrete values
$V_{LSB} = V_{REF}/2^N$
$V_{OUT} = V_{LSB} * (\text{Input code})_d = (V_{REF}/2^N) * (\text{Input code})_d$
V_{OUT} (full-scale) = $V_{LSB} * (2^N-1) = V_{REF} - V_{LSB}$
Output impedance is equal to R, regardless of the number of bits, or the binary input code
two resistor values facilitates fabrication and integration

V_{REF} is applied to the R-2R inputs in a binary-encoded manner that is, binary levels, a_X=0 or a_X= V_{REF}

Binary-Weighted DAC Inputs

V_{REF} @ a_0:	$V_{LSB} * 2^0 = 1 * V_{LSB}$	OR $V_{REF}/2^N = V_{REF}/16$	LSB
V_{REF} @ a_1:	$V_{LSB} * 2^1 = 2 * V_{LSB}$	OR $V_{REF}/2^{N-1} = V_{REF}/8$	
V_{REF} @ a_2:	$V_{LSB} * 2^2 = 4 * V_{LSB}$	OR $V_{REF}/2^{N-2} = V_{REF}/4$	
V_{REF} @ a_3:	$V_{LSB} * 2^3 = 8 * V_{LSB}$	OR $V_{REF}/2^{N-3} = V_{REF}/2$	MSB
0V applied to any input contributes		0V	

Inputs are additive to V_{OUT} and independent of the others.
$V_{OUT} = V_{LSB} * (\text{Input code})_d = (V_{REF}/2^N) * (\text{Input code})_d$



4-bit Example
If V_{REF} = 16V and assuming the following binary-encoded values, $a_{(3-0)}$, what is V_{OUT}

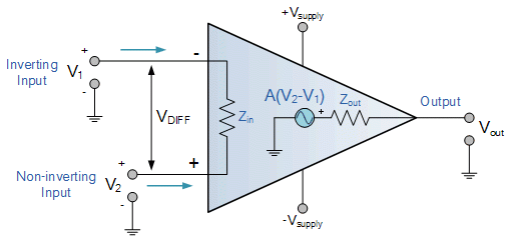
$a_{(3-0)} = 0xC = 1100_b$	$V_{OUT} = 8*V_{LSB} + 4*V_{LSB} = 8*1V + 4*1V$	$= 12V$
$a_{(3-0)} = 0x8 = 1000_b$	$V_{OUT} = 8*V_{LSB} = 8*1V$	$= 8V$
$a_{(3-0)} = 0x3 = 0011_b$	$V_{OUT} = 2*V_{LSB} + 1*V_{LSB} = 2*1V + 1*1V$	$= 3V$

Using $V_{OUT} = V_{LSB} * (\text{Input code})_d$

$a_{(3-0)} = 0xC = 1100_b$	$V_{OUT} = 1V * (0xC)_d = 1V * 12$	$= 12V$
$a_{(3-0)} = 0x8 = 1000_b$	$V_{OUT} = 1V * (0x8)_d = 1V * 8$	$= 8V$
$a_{(3-0)} = 0x3 = 0011_b$	$V_{OUT} = 1V * (0x3)_d = 1V * 3$	$= 3V$

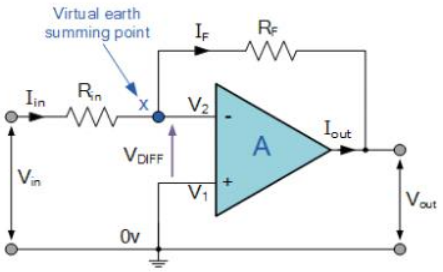
OpAmp Topics

The operational amplifier is a high gain difference amplifier. $V_O = A_V (V_+ - V_-)$



Differential Input	$V_+ - V_-$
Open-loop gain	High, A_V ~ 100,000
Input impedance	High, ~100KΩ to >1GΩ
Open-loop Output Z	Low, varies, maybe 100Ω
Assume: infinite input Z	zero output Z

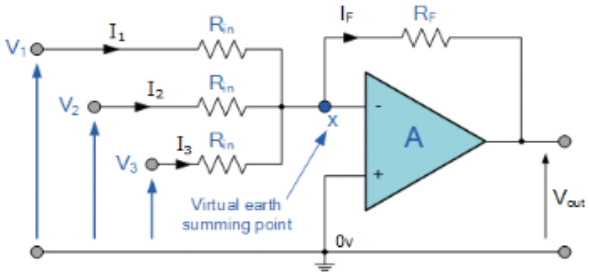
Inverting Amp



Circuit gain, A_V $V_O / V_{IN} = -(R_F / R_{IN})$

$V_O = -V_{IN} * (R_F/R_{IN}) = -V_{IN} * A_V$

Inverting Summing Amp



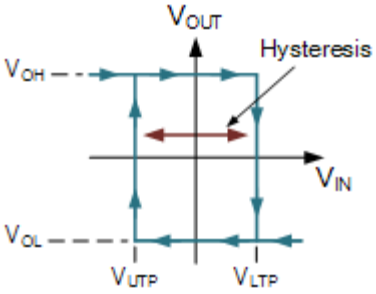
$V_O = -(V_1A_{V1} + V_2A_{V2} + V_3A_{V3})$

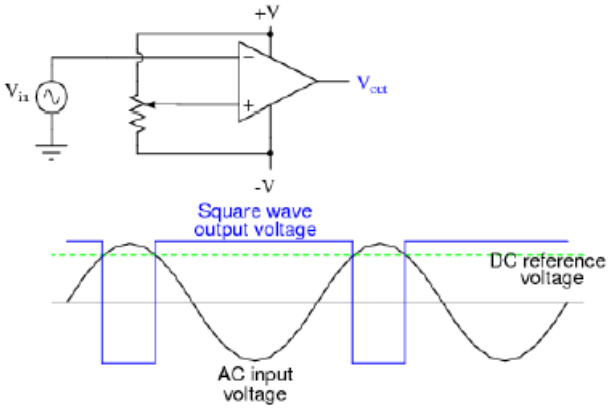
IF $V_1=V_2=V_3= V_{IN}$,

$V_O = -V_{IN} * (R_F/R_{In1} + R_F/R_{In2} + R_F/R_{In3}) = -V_{IN} * (A_{V1} + A_{V2} + A_{V3})$

Comparator

The comparator output switches, $+V_{CC}$ to $-V_{CC}$, in the direction of the larger input voltage.





Inverting Integrator (aka Ramp Generator)

A step voltage at the input will generate a ramp voltage at the output

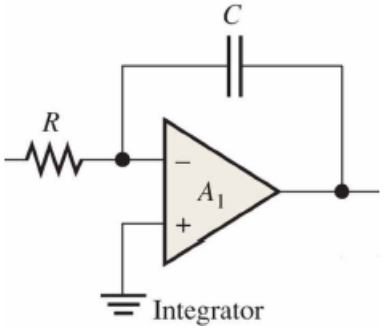
Capacitor voltage equals charge (Q Coulombs) divided by capacitance (C Farads)

that is 1V=1Q/1F or $V_{CAP} = Q / C$

Current (I) is defined as 1Amp = 1Q/1Sec (ie charge flow per time)

So... 1Q = 1Amp*1Sec or $Q = I * t$

Substitue $I * t$, $V_{CAP} = I * t / C$



For the Inv Integrator let $R = 1\Omega$ $C = 1F$ $V_{IN} = 1V$ $V_{CAP} = V_{OUT}$ $I_{IN} = V_{IN} / R$

$V_{OUT} = -(V_{IN} / R) * t / C$ (V_{OUT} after t sec with V_{IN} applied)

$t = -V_{OUT} * C / (V_{IN} / R)$ (time required for V_{OUT} to ramp to V_{IN})

Example: 4-bit Binary-Weighted D/A Converter

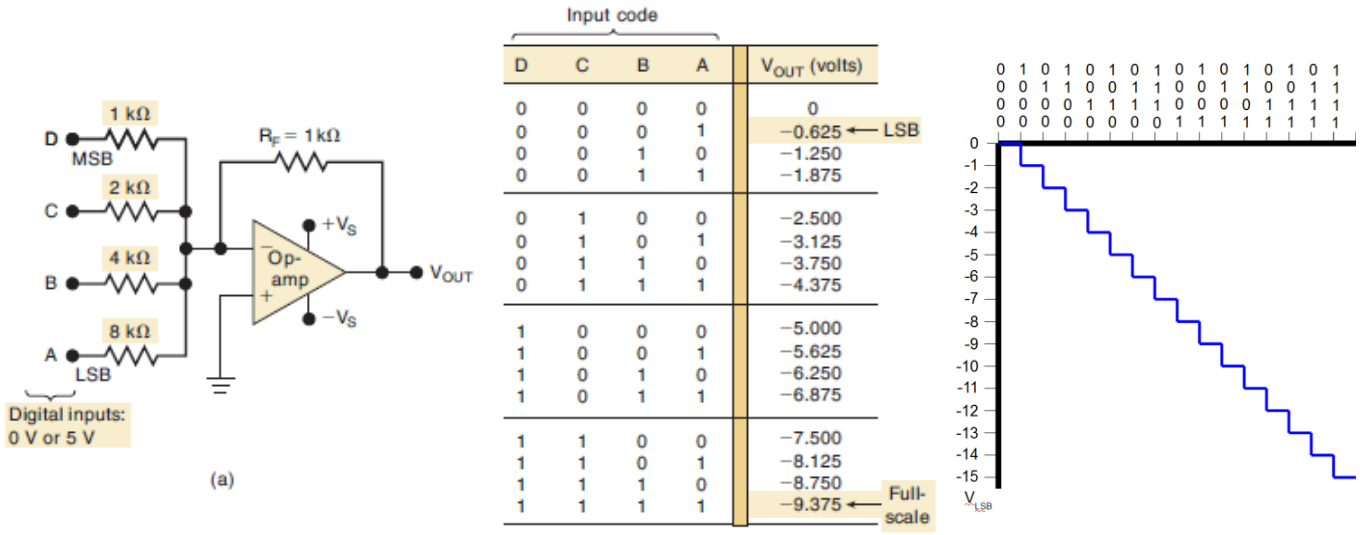


Fig 11.5 Ref Digital Systems textbook Ch 11.3 DAC Circuitry

4-bit DAC with binary-weighted input resistors & Vref = 5V

Av LSB = - 1/8 ….. Av MSB = - 1

$V_O = -(V_{REF}A_{VA} + V_{REF}A_{VB} + V_{REF}A_{VC} + V_{REF}A_{VD})$

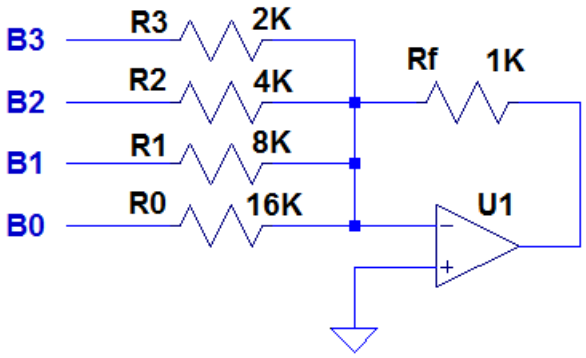
$V_O = -V_{REF}*(A_{VA}[B_A] + V_{REF}A_{VB}[B_B] + V_{REF}A_{VC}[B_C] + V_{REF}A_{VD}[B_D])$

$V_{LSB} = V_{REF}*(- 1/8) = -0.625V$

$V_{MSB} = V_{REF}*(- 1) = -5V$

$V_{FS} = -5V - 2.5V - 1.25V – 0.625V = -9.375V$

For this Binary-Weighted DAC:



V_{REF} = 5V

$V_O = -V_{REF}*(A_{V2}B_3 + A_{V2}B_2 + A_{V1}B_1 + A_{V0}B_0)$

What is V_{LSB}?

$V_{LSB} = - 5V*[0 + 0 + 0 + (1/16)] = -0.3125V$

What is V_{OUT} when the V_{REF} input is binary-encoded as 0x8 (1000b)?

$V_{OUT} = - 5V*((1/2) + 0 + 0 + 0) = -2.5V$

What is V_{OUT} when the V_{REF} input is binary-encoded as 0xC (1100b)?

$V_{OUT} = - 5V*((1/2) + (1/4) + 0 + 0) = -3.75V$

What is V_{FS} ?

$V_{FS} = - 5V*((1/2) + (1/4) + (1/8) + (1/16) = -4.6875V$

Limited resolution:

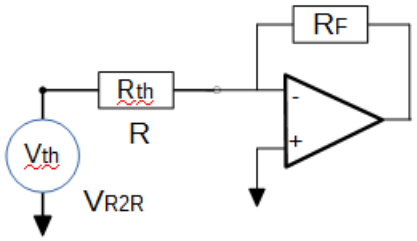
Resistor value range can span several orders of magnitude for medium resolution.
At 12-bit, if R_{MSB} = 1KΩ R_{LSB} = 2MΩ

The fabrication of precision IC resistances over a large range of values is problematic - difficult to maintain an accurate R ratio over temperature variation.

Example: 4-bit R-2R D/A Converter

Thevenin Equivalent of the R2R DAC

https://www.best-microcontroller-projects.com/R-2R-ladder.html



Thevenin equivalent circuit gain :

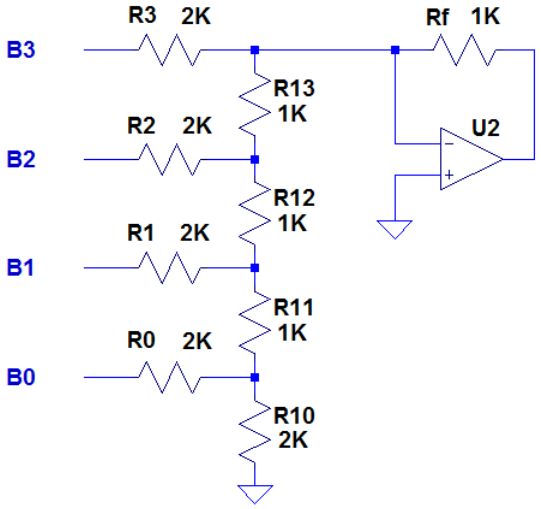
$A_V = - R_F/R_{TH}$

Thevenin equivalent V_{R2R} :

$V_{TH} = V_{LSB} * (\text{Input Code})_d$

$= (V_{REF}/2^N) * (\text{Input Code})_d$

Example R2R DAC: $V_{REF} = 5V$



What is V_{OUT} when the V_{REF} input is binary-encoded as 0x8 (1000b)?

$V_{OUT} = A_{VTH} * V_{TH}$

$= -(R_F/R_{TH})*(V_{REF}/2^N)*(\text{Input Code})_d$

$= (-1)*(5V/16)*(8) = -2.5V$

What is V_{OUT} when the V_{REF} input is binary-encoded as 0xC (1100b)?

$V_{OUT} = (-1)*(5V/16)*(12) = -3.75V$

What is V_{FS} ?

$= (-1)*(5V/16)*(15) = -4.6875V$

4-bit R2R Ladder binary-weighted input:

V_{REF} @ B_3 $V_{OUT} = 8V_{LSB}$ OR $V_{REF}/2$

V_{REF} @ B_2 $V_{OUT} = 4V_{LSB}$ OR $V_{REF}/4$

V_{REF} @ B_1 $V_{OUT} = 2V_{LSB}$ OR $V_{REF}/8$

V_{REF} @ B_0 $V_{OUT} = 1V_{LSB}$ OR $V_{REF}/16$

Memory Topics

Volatile mem – SRAM, DRAM

Volatile memory retains memory only when power to the memory is on.

Static – retains data indefinitely with power on.

Dynamic – retains data temporarily with power on.

Simpler cells that NVM, 6T-4T-1T1C smaller, faster, less expensive, used for the main memory in most computers.

Non-Volatile mem – ROM, PROM, EPROM, EEPROM, Flash

NVM Retains memory when power to the memory is off.

Non-volatile semiconductor memory (NVSM, since EPROM) stores data in floating-gate MOSFET memory cells.

ROM – fixed memory, no write capability, photo-lithographic mask during IC fab

PROM – one-time programmable (OTP), fixed thereafter , originally fuse-based technology

EPROM – programmable, UV erasable - complete memory array only

EEPROM – programmable, electrically erasable - any individual addressable location - via control logic & prog voltage

Flash – >10 yr retention, finite WR endurance up to 1M, WR/Erase of blocks/sections only

Memory Access Methods

Methods to access memory locations:

1. **Random Access**: each memory location is uniquely accessible in any order and in uniform time.
2. **Sequential Access**: memory access in sequential order - magnetic or optical disk HDs.
3. ~~**Direct Access**: Information is stored in tracks, with each track having a separate read/write head - mag/opt/SSD drives.~~

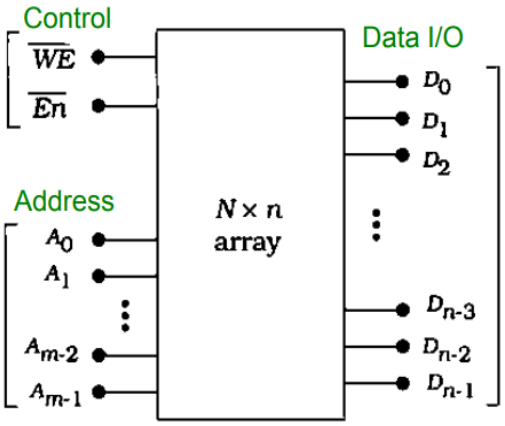
EET123 will focus on Random Access

Memory Access - Addressability

Address decoding logic will organize the cell array as N x n

N addressable locations - row logic will uniquely address each data group

n bit depth - bytes, 16-bit words, 32-bit words, etc



Memory N x n organization

1Kx8:	8-bit data	1024 addr locations	8K bits total	10-bit addr
1Kx16:	16-bit data	1024 addr locations	16K bits total	10-bit addr
16Kx8:	8-bit data	16384 addr locations	128K bits total	14-bit addr
1Mx8:	8-bit data	1048576 addr locations	8M bits total	20-bit addr

Memory Cells

SRAM Cell 6T, 4T

Larger/ costs more than DRAM, faster than NVRAM

DRAM Cell 1T1C

Smaller/costs less than SRAM, slower than SRAM, requires refresh

EEPROM Cell Floating-gate MOSFET

Programming techniques, hot electron, Fowler–Nordheim emission

Rd/Wr to individual address location – slower than flash

Flash Cell Floating-gate MOSFET

Rd to individual address location

Wr/Erase large block address locations (sectors) – faster that EEPROM

Harvard Arch vs Von Neumann Arch

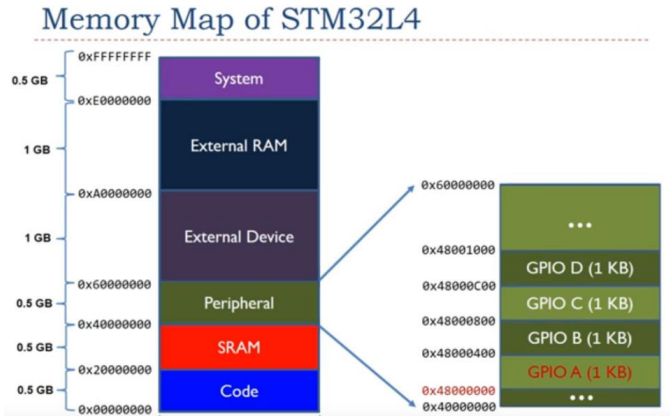
The **Von Neumann architecture** exhibits one addr/data bus for both instruction and data memory.

The **Harvard architecture** exhibits one addr/data bus for program memory and a separate addr/data bus for data memory, allowing simultaneous access to both instructions and data.

The **modified Harvard architecture** is a variation of the Harvard computer architecture that, unlike the pure Harvard architecture, allows the contents of the instruction memory to be accessed as data. Most modern computers that are documented as Harvard architecture are, in fact, modified Harvard architecture.

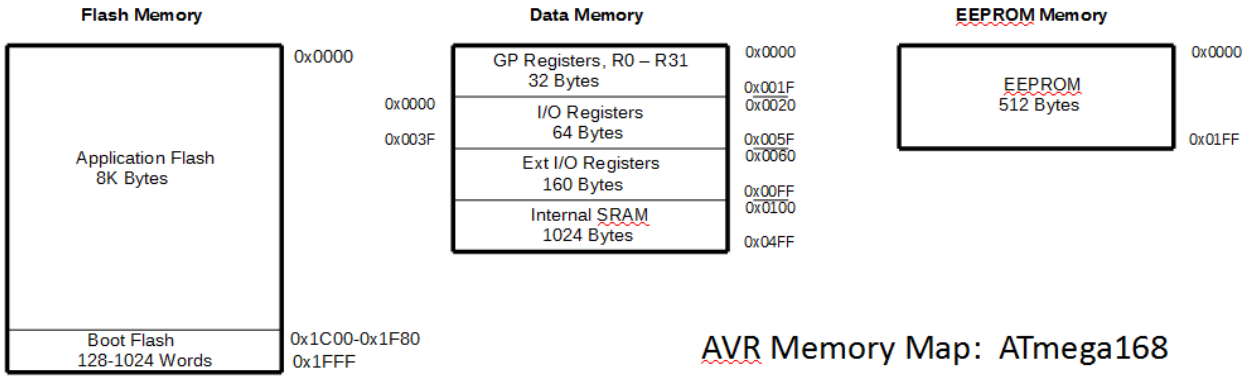
Von Neumann memory address space

All addressable locations reside within one contiguous address space.



AVR microcontrollers use the Modified Harvard architecture.
Separate memory types connect to separate buses.

Program memory	Non-volatile Flash
Data memory	Volatile SRAM Data
EEPROM memory	Non-volatile EEPROM Data



AVR Memory Map: ATmega168

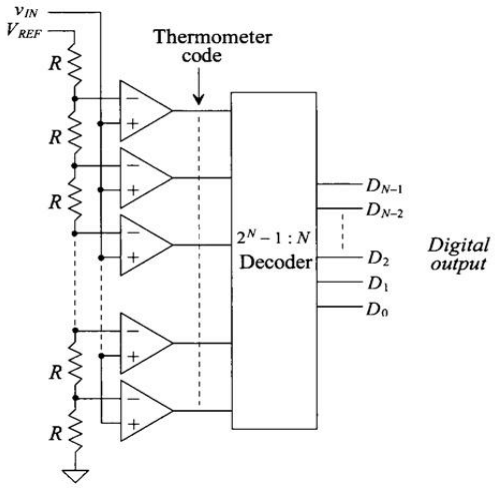
Access is encoded by commands with address operands that uniquely identify specific memory locations.
So far, we have used several commands to access the SRAM register locations, GP Regs, I/O Regs, Intern SRAM locations:

GP Register Commands	
clr Rd	clear register
dec Rd	decrement register
inc Rd	increment register
ser Rd	set register
mov Rd, Rr	copy register

Commands for Upper SRAM Addr Locations	
ldi Rd, K	Load immediate data K into GP Reg
ld Rd, k	Load indirect from addr k into GP Reg
sts k, Rr	store direct to data addr k from GP Reg
st X, Rr	store indirect to data addr within X from GP Reg

ADC Types

Flash ADC

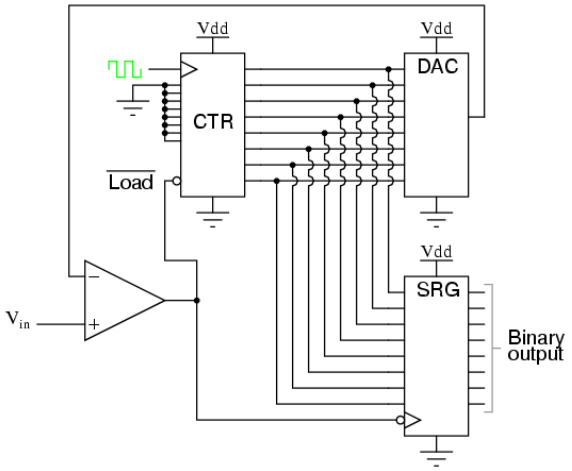


For an N-bit Flash ADC
requires 2^N resistors in a string ladder resistor divider
requires 2^N-1 comparators
requires 2^N to N priority encoder logic with 'bubble error' detect/correct
- 8 to 3 priority encoder for 3-bit conversion
- 16 to 4 priority encoder for 4-bit conversion
- 256 to 8 priority encoder for 8-bit conversion

Several techniques to detect/correct 'bubble error'
> 8bits decreasingly practical,

Pos fastest ADC - maybe 10 nS delay time, simple concept
Neg But requires 2^N-1 comp & 2^N res which limit resolution

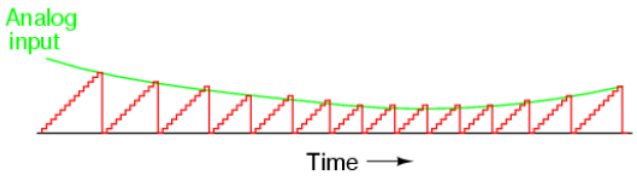
Digital Ramp ADC



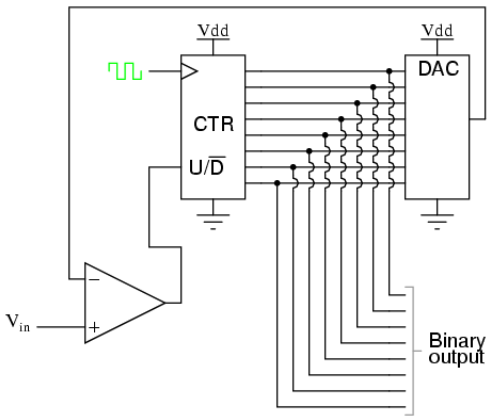
Counter up-counts
DAC stair-steps V_{DAC} until it reaches the V_{IN} threashold.
Then the counter resets, V_{DAC} goes to zero, conversion starts again

T_{CONV} of each sample $= (V_{IN}/V_{LSB}) * t_{CLK} = V_{IN}/(V_{REF}/2^N) * t_{CLK}$
$= 2^N * (V_{IN}/V_{REF}) * t_{CLK}$

Pos simple design concept, low cost implementation
clk can decrease conversion time, typ faster than dual slope
Neg But still a slower technique, requires DAC, variable conversion time,

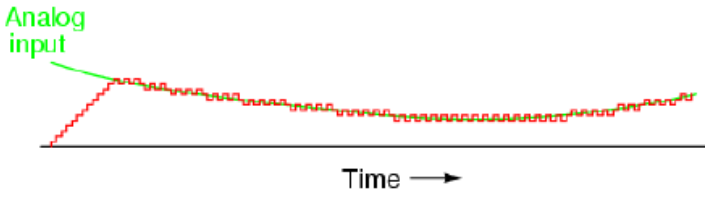


Tracking ADC



Similar to the Dig Ramp ADC, uses U/D counter
Faster sustained conversion, initial conv ramp is visible at the output,
a conversion every clk is possible for slow signals,
exhibits bit bobble at a constant Vin

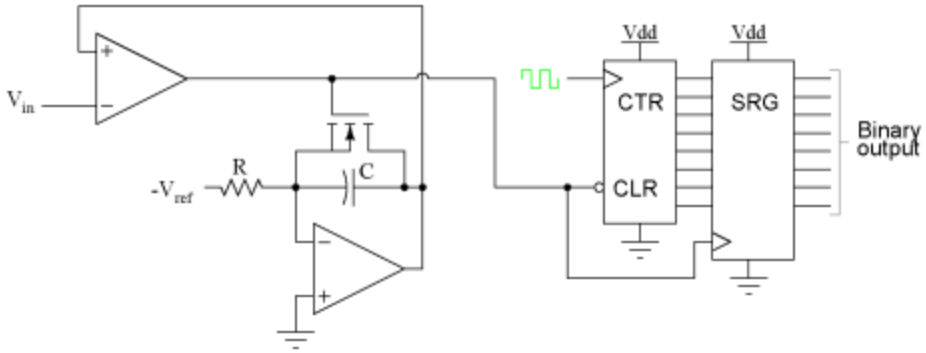
Pos faster conv after init ramp, 1 clk updates possible, simple concept, clk controls conver time
Neg but requires DAC, output dithers 1LSB with a stable input



Single-Slope ADC

Similar to the Dig Ramp ADC, replaces the DAC with an Inv Integrator, conversion time = integration time

Pos No DAC, simple design, low cost, clk can decrease conver time
Neg But still a slower technique, variable conversion time, must be calibrated - integ time must = FS cnt time, and analog component values may drift

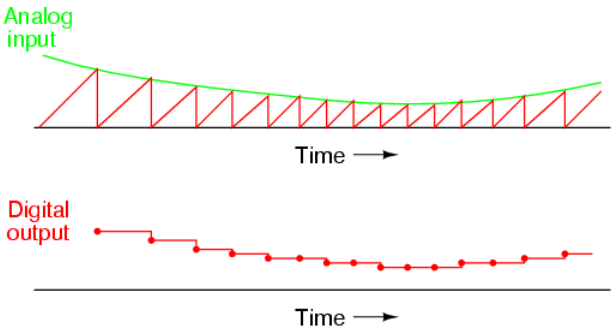


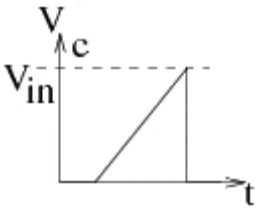
What is the integ time? (t_{INT}=t_{CONV})

Remember V=Q/C ?
$V_{CAP} = I*t\ /C$ (sub I *t for Q)
$V_{OUT} = -(V_{REF}/R)*t\ /C$ (sub V_{REF}/R for I)

$t = -V_{OUT}*R*C/V_{REF}$ (solve for t)
$t = -V_{IN}*R*C/V_{REF}$ (V_{OUT} =V_{IN} @ EOC)

Time required for the integrator V_{OUT} to reach V_{IN} is the conversion time.

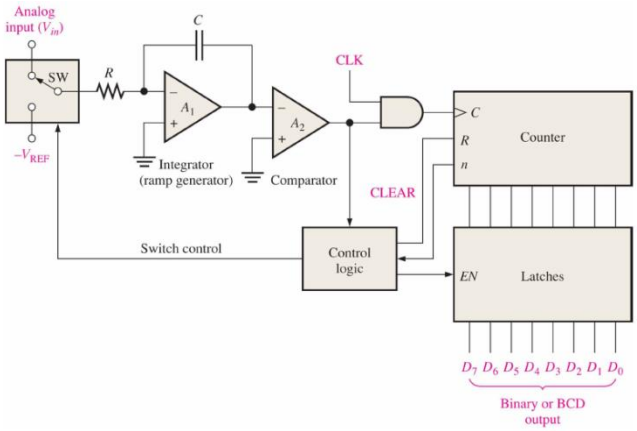


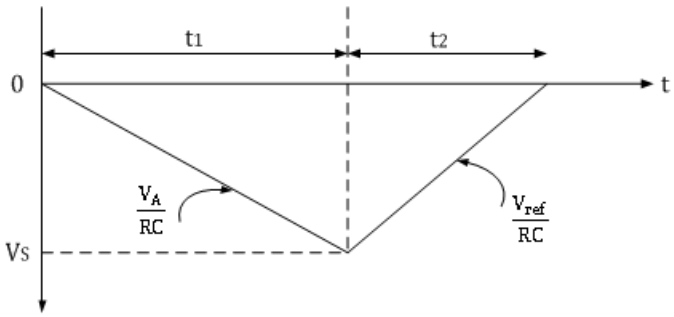


Dual-Slope ADC

$V_{IN}*t_{INT} = V_{REF}*t_{DE\text{-}INT}$ $T_{CONV} = t_{INT} + t_{DE\text{-}INT}$ $t_{DE\text{-}INT} = t_{CLK} * (\text{Cntr Out})_d$ (dec equiv of counter output)
Slow … but, averaging reduces noise, good linearity, no drift,
high resolution achievable, good accuracy, immune to comp drift

Pos No DAC, simple design, low cost, clk can decrease conver time, no drift error
Neg But still a slower technique, variable conversion time



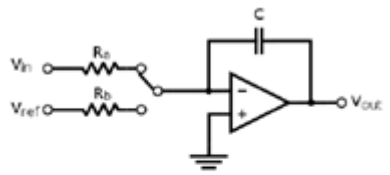


$0 = V_{OUT1} + V_{OUT2}$
$= -(V_{IN}/R)*t_1/C + [-(V_{REF}/R)*t_2/C]$

$\Rightarrow (V_{IN}/R)*t_1/C = -(V_{REF}/R)*t_2/C$
$(V_{IN}\ \cancel{/R})*t_1\ \cancel{/C} = -(V_{REF}\cancel{/R})*t_2\ \cancel{/C}$

$V_{IN}*t_1 = |V_{REF}|*t_2$

$V_A*t_{INT} = |V_{REF}|*t_{DE\text{-}INT}$ $V_A/V_{REF} = t_2/t_1$ $T_{CONV} = t_{INT} + t_{DE\text{-}INT}$



Enhanced Dual-Slope ADC

$V_S = (V_A/R_A)*t_1/C = (V_{REF}/R_B)*t_2/C$
$(V_A/R_A)*t_1 = (V_{REF}/R_B)*t_2$

Example

For an analog V_{IN} = 2.5V and V_{REF} = 5V and t_{INT} = 10mS What is the total conversion time?

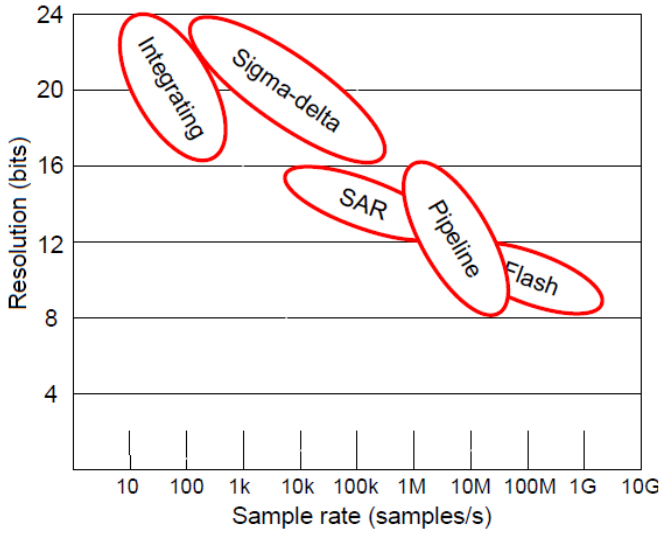
$t_{DE\text{-}INT} = (V_{IN}/|V_{REF}|)*t_{INT}$

Total Conv time = $t_{INT} + t_{DE\text{-}INT}$ = 10mS + 10mS*(2.5/5) = 10mS + 5mS = 15mS

What is the Counter output?

Cntr Out = $(t_{DE\text{-}INT}/t_{CLK})_b$ (Counter Out is the binary equiv of the decimal ratio $t_{DE\text{-}INT}/t_{CLK}$)

ADC Performance Comparison: LECTURE-AD_Conversion_Part2_Wk2 Pg 42



- Flash (or Simultaneous)
- (where are Digital Ramp and Tracking?)
- Integrating (Single-slope and Dual-slope)
- SAR (Successive approximation register)
- Pipeline (Subranging or Half-flash)
- Sigma-delta