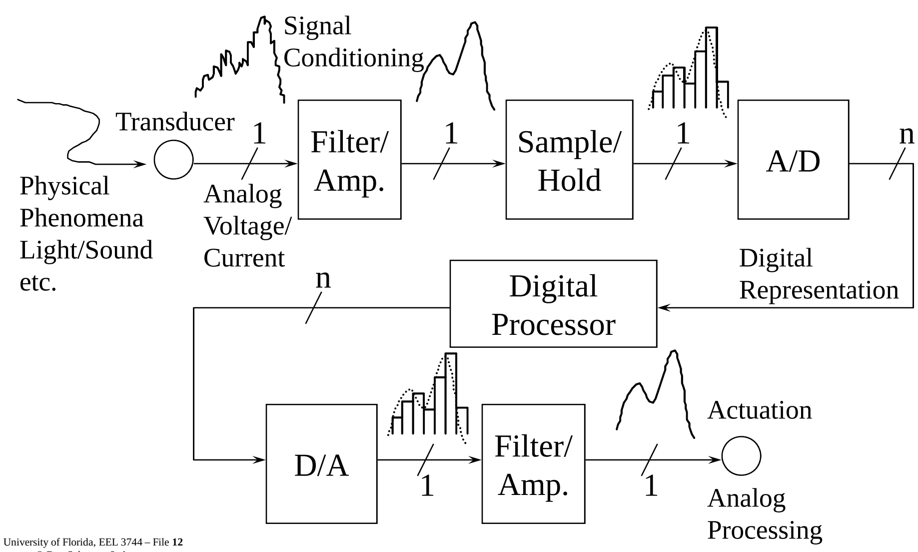
EEL 3744

Menu

- A/D-D/A Conversion Processes
  - Example: "*Grandma Singing Hymns*"
- Digital Signal Processing
- Analog-to-Digital Conversion
  - A/D Conversion Methods
- Operational Amplifier in D/A & A/D
- Digital-to-Analog

EEL 3744

A/D and D/A Conversion Process



EEL 3744

DSP (Digital Signal Processing) in ECE

- Discussion on Digital Signal Processing
  > Analog filters introduced in (*EEL 3112*)
  > Analog Filter Design (*EEL 3308: Electronic Circuits 1*), i.e., s-plane design
  > Digital Analysis of Signals (*EEL 3135: Intro to [Discrete-Time] Signals & Systems*), i.e., z-plane design
  > Digital Filtering (Analog-->Digital) (*EEL 4712: Digital Design,* ***<u>4511</u>: Real-time DSP Applications****, 4750: Intro to DSP, EEE 5502: Foundations of Digital Signal Processing, 6503: Digital Filtering*)
  > Implementation as a Recursive Program (CE Stuff)
  > Advantages of Digital Signal Processing (*EEL 4511 & 4750*)
- The use of D-A/A-D and Digital Processors allows computer engineers to create digital solutions to many engineering problems

EEL 3744

Programming Example

- Example: Recovery of "*Grandma Singing Hymns*" cassette tapes from the 1950's
  > High-pass filter to remove low frequency tape hum (there was no Dolby system in the 1950's)
  > Low pass filter to remove high frequency media damage (hiss)
  > What would it take to do this with Analog circuits?
  > What would it take to do this with Digital technology?
- The Analog Filter
  $H(s) = Y(s)/X(s) = 1/(s^2+bs+c)$

EEL 3744

Programming Example

- The Analog Filter
  $H(s) = Y(s)/X(s) = 1/ (s^2+bs+c)$
- The Digital Filter
  $H(z) = Y(z)/X(z) = 1/ (z^2+\alpha z+\beta)$
- This results in the **difference equation**:
  $Y(z) = X(z) / (z^2+\alpha z+\beta)$
  $X(z) = Y(z) (z^2+\alpha z+\beta)$
  $X(z) = z^2 Y(z) + \alpha zY(z)+ \beta Y(z)$ or
  $z^{-2} X(z) = Y(z) + \alpha z^{-1}Y(z)+ \beta z^{-2}Y(z)$ or
  $X(n-2) = Y(n) + \alpha Y(n-1) + \beta Y(n-2)$ or
  $Y(n) = -\alpha Y(n-1) - \beta Y(n-2) + X(n-2)$

EEL 3744

Programming Example

- This results in the program (say in Fortran):

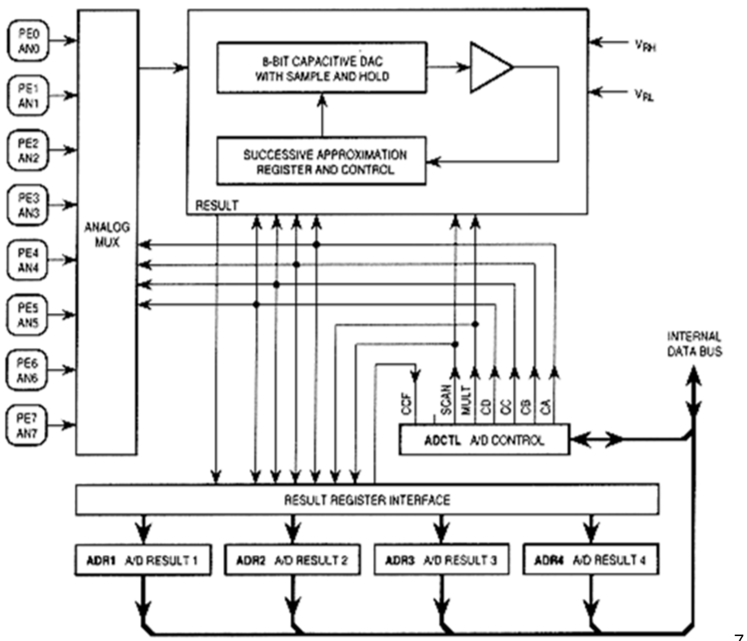
```
C       Y(n) = -αY(n-1) - βY(n-2) + X(n-2)
        Data n/__ /
        DIMENSION Y(0..n), X(0..n)
        READ (5,x) X,alpha,beta,Y(0),Y(1)
        DO 100 i=2,n
100     Y(i) = X(i-2) - (alpha*Y(i-1) + beta*Y(i-2))
```

- Thus, given the parameters α, β and inputs X(0..n) we can calculate the Y(2..n) if we know Y(0) and Y(1)
- This program can be implemented easily(?) in any (e.g., XMEGA) assembly language

EEL 3744

68HC11
A/D Block
Diagram

TD:Fig.10-1



EEL 3744

Dual slope (Integrator) A/D Converters

- **Components:** Use a counter, analog integrator, voltage comparator, reference voltage and control circuitry
- **Technique:** Integrate over one cycle of the power line frequency (60 Hz) and therefore can ignore 60 Hz noise
- **High points:** Accuracy good, 60 Hz noise immunity
- **Low points:** Relatively slow

EEL 3744

Flash (Simultaneous) A/D Converters

- **Technique & components:** Uses a tapped resistor to divide a reference voltage into 2^n equal parts, 2^n voltage comparators (to compare the input voltage against each of the tap voltages, and a priority encoder (2^n inputs and n-bit coded output) to output the digital signal
- **High points:** Accuracy very good if use high precision resistors; very fast (fastest); speed depends only on propagation delays (mostly in the encoder)
- **Low points:** Requires a **tremendous** amount of circuitry

EEL 3744

Sigma-Delta A/D Converters

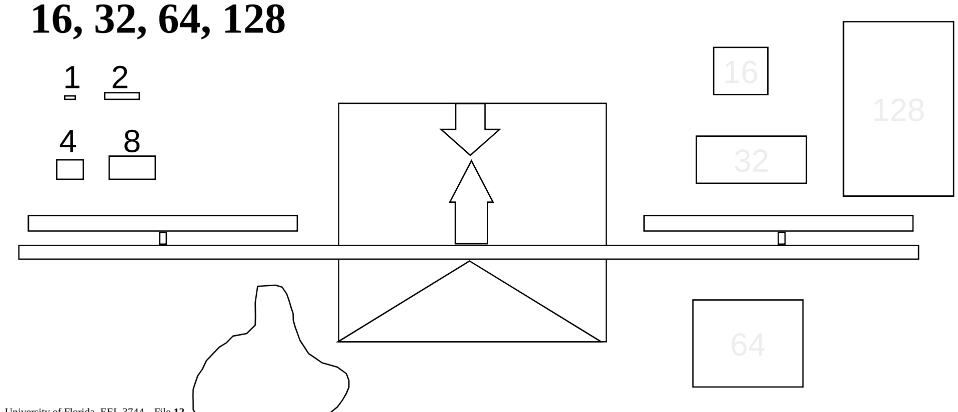
- **Technique & components:** Uses high speed sampling a digital decimation filter. (See class in digital filters.)
- **High points:** Accuracy good
- **Low points:** May be slow
- **Note:** This is now common in many processors (including XMEGA and ATMega)

EEL 3744
Successive-Approximation A/D Converters

- **Technique & components:** Uses a D/A converter (DAC), voltage comparator, reference voltage and control circuitry
- **High points:** Relatively fast (not as fast as flash, but much faster than dual-slope)
- **Low points:** Requires a DAC
- **Note:** This has been common in many processors (including 68HC11, 68HC12, TMS320 DSC)

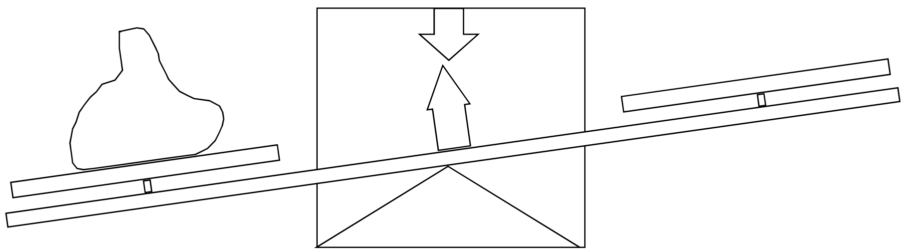
EEL 3744
Successive-Approximation A/D Converter Example

- **Given: Ancient balance scale, some "boxes" of sand of <u>known</u> weight (in *talanton* units): 1, 2, 4, 8, 16, 32, 64, 128**



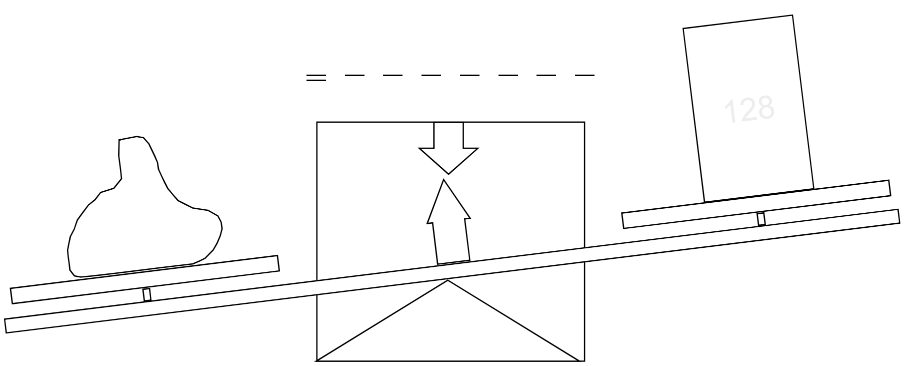
EEL 3744
Successive-Approximation A/D Converter Example

- **Find the weight of a sack of salt (worth its weight in gold).**



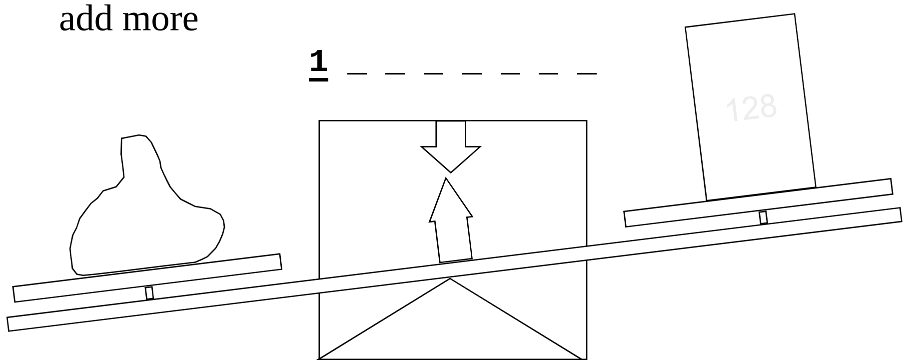
EEL 3744
Successive-Approximation A/D Converter Example

- **First try the maximum weight (128 talatons)**



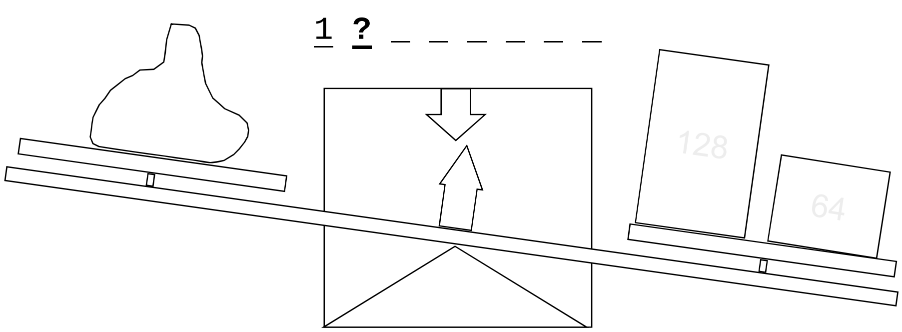
EEL 3744
Successive-Approximation A/D Converter Example

- **First try the maximum weight (128 talatons)**
  > Too light, so keep this weight (put 1 in the MSB) and add more



EEL 3744
Successive-Approximation A/D Converter Example

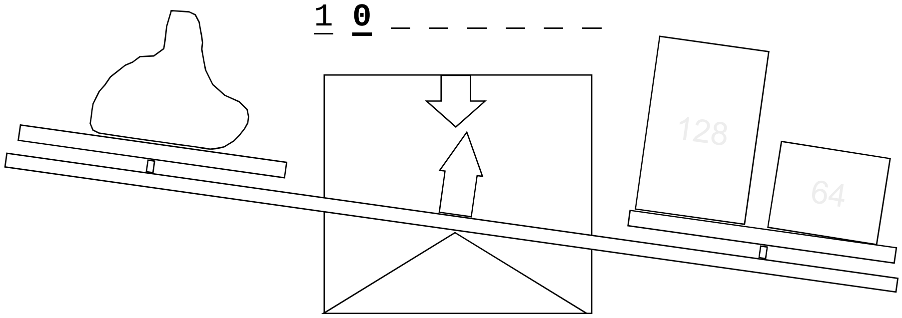
- **Now try the next weight down (64 talatons)**



EEL 3744
Successive-Approximation A/D Converter Example

- **Now try the next weight down (64 talatons)**
  > Too heavy, so remove the 64 talaton weight and put a 0 below

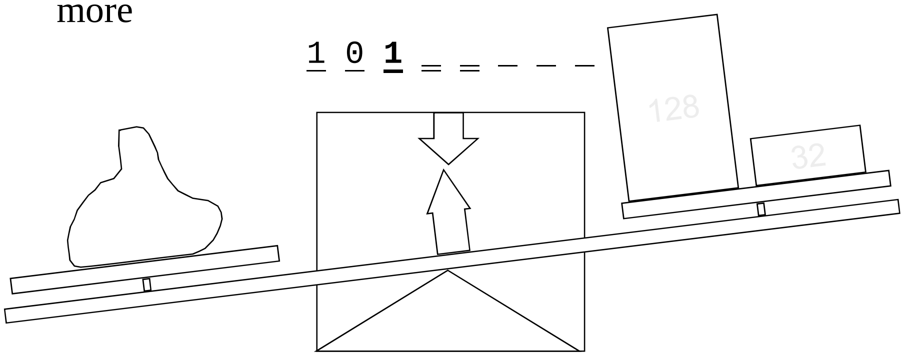
1 **0** _ _ _ _ _ _



EEL 3744
Successive-Approximation A/D Converter Example

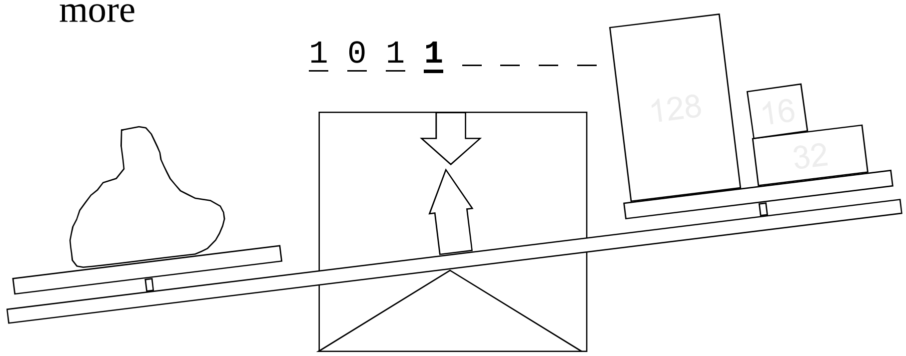
- **Try 32 talaton weight**
  > Still too light, so keep this weight (put 1 below) and add more

1 0 **1** _ _ _ _ _



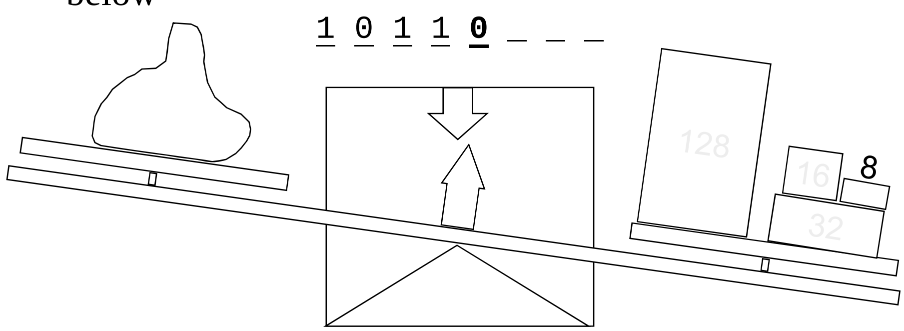
EEL 3744
Successive-Approximation A/D Converter Example

- **Try 16 talaton weight**
  - > Still too light, so keep this weight (put 1 below) and add more



EEL 3744
Successive-Approximation A/D Converter Example

- **Now try the 8 talaton weight**
  - > Too heavy, so remove the 8 talaton weight and put a 0 below

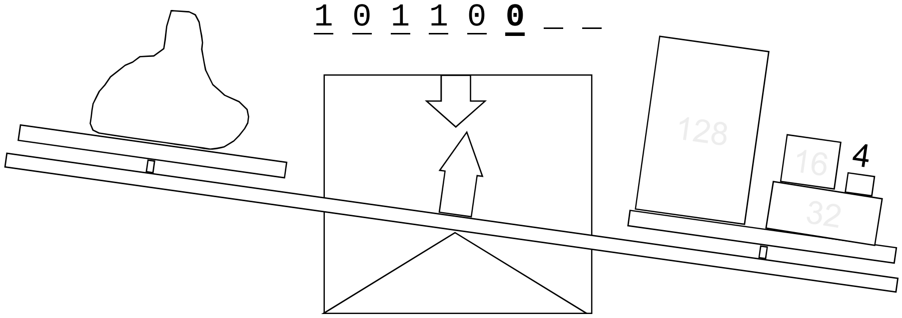


EEL 3744

Successive-Approximation A/D Converter Example

- **Now try the 4 talaton weight**
  - > Too heavy, so remove the 4 talaton weight and put a 0 below

1 0 1 1 0 **0** _ _

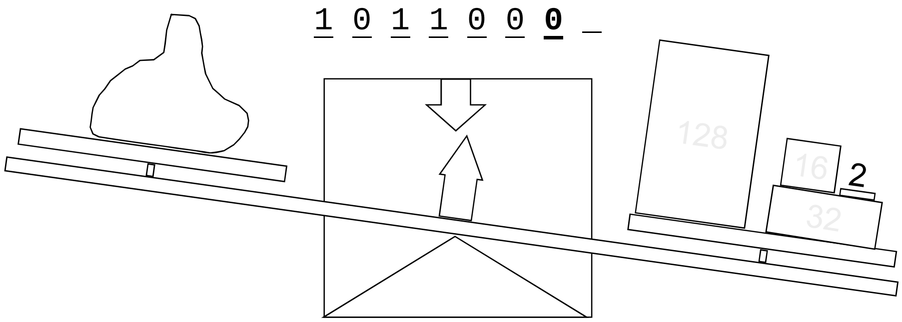


EEL 3744

Successive-Approximation A/D Converter Example

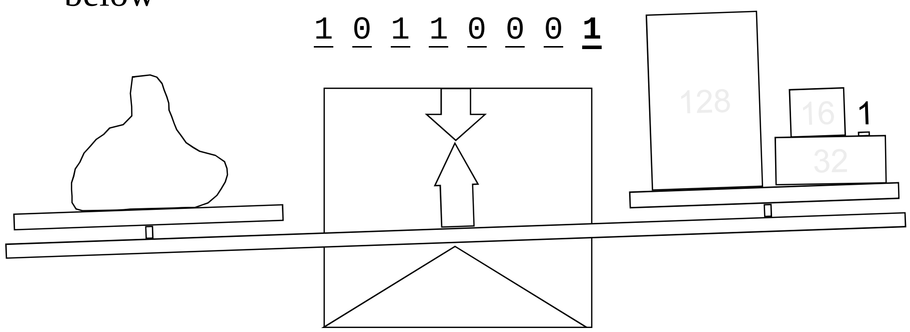
- **Now try the 2 talaton weight**
  - > Too heavy, so remove the 2 talaton weight and put a 0 below

1 0 1 1 0 0 **0** _



EEL 3744
Successive-Approximation A/D Converter Example

- **Now try the 1 talaton weight**
  > Not quite heavy enough, so keep this weight and put a 1 below



Q: What is the range of **possible** weights for **this** bag?

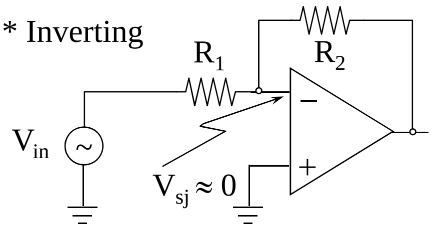
EEL 3744
Successive-Approximation A/D Converter

- The above algorithm is called a **binary search** (since the know weights were $\mathbf{2^n}$ talatons)
- The electronic successive-approximation A/D works with the same principles as the ancient balance, comparing the unknown quantity (voltage) with a succession of know quantities (voltages) with binary weights
- All Successive-Approximation A/D converters have D/A converters (DACs) inside

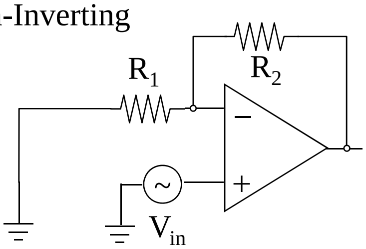
EEL 3744

OP Amps Circuits used in A/D & D/A Circuits

* Inverting



* Non-Inverting



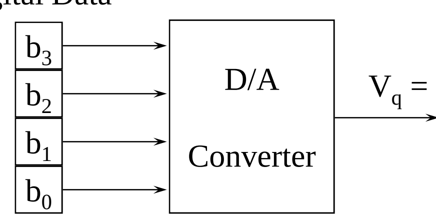
EEL 3744

Digital/Analog Conversion

Binary integer: $k = b_3\ 2^3 + b_2\ 2^2 + b_1\ 2^1 + b_0$

Digital Data

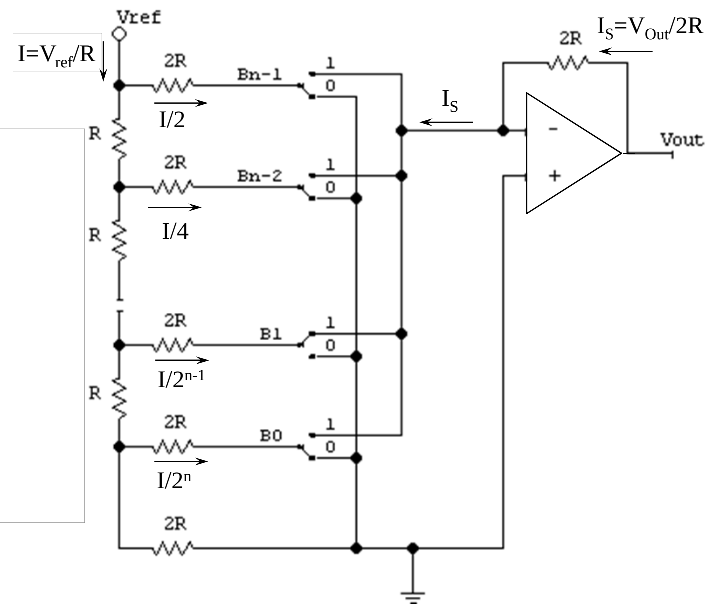
Register



$V_q = k\ \Delta V =$ quantized analog voltage

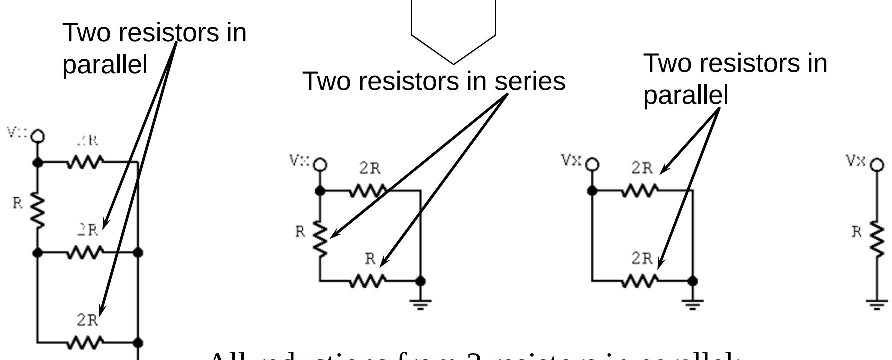
$\Delta V =$ quantized voltage proportionality constant

EEL 3744 DAC: Inverted R-2R Ladder Circuit



EEL 3744
DAC: Inverted R-2R Ladder Circuit

Demonstration that equivalent resistance of network is **always** R



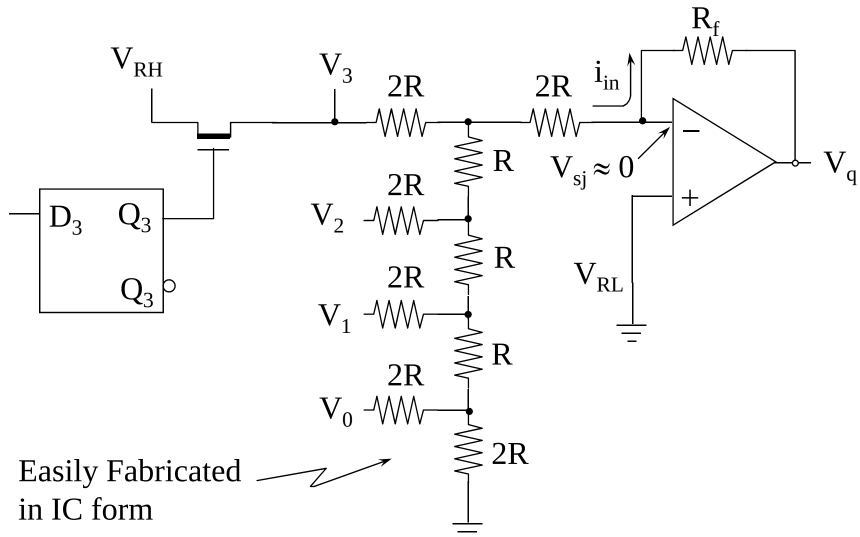
All reductions from 2 resistors in parallel:

$R_p = R_1 R_2/(R_1 + R_2)$

Or two resistors in series: $R_s = R_1 + R_2$

EEL 3744

DAC: R-2R Ladder Network

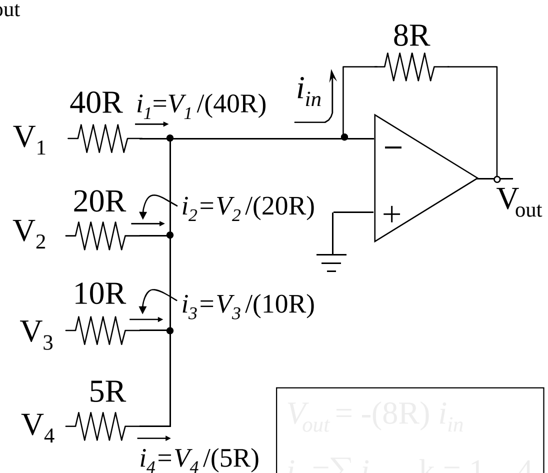


EEL 3744

Another 4-Bit DAC

All table entries in Volts

V_4	V_3	V_2	V_1	$-V_{out}$
0	0	0	0	0
0	0	0	5	1
0	0	5	0	2
0	0	5	5	3
0	5	0	0	4
0	5	0	5	5
0	5	5	0	6
0	5	5	5	7
5	0	0	0	8
5	0	0	5	9
		⋮		
5	5	0	5	13
5	5	5	0	14
5	5	5	5	15



$V_{out} = -(8R)\, i_{in}$

$i_{in} = \sum i_k$, k = 1…4

V_k = 0V or 5V

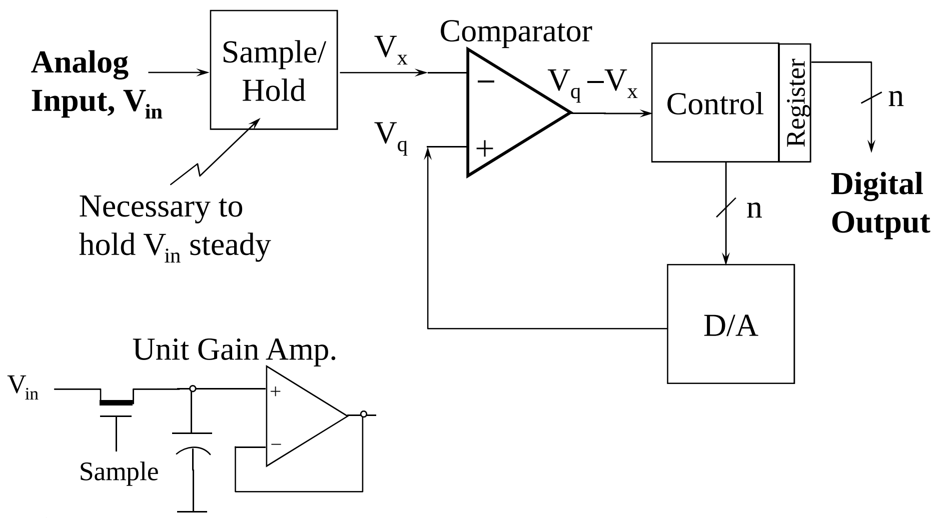
EEL 3744

Analog/Digital Conversion

- A/D Conversion Methods
  - >Successive Approximation
  - >Flash Conversion

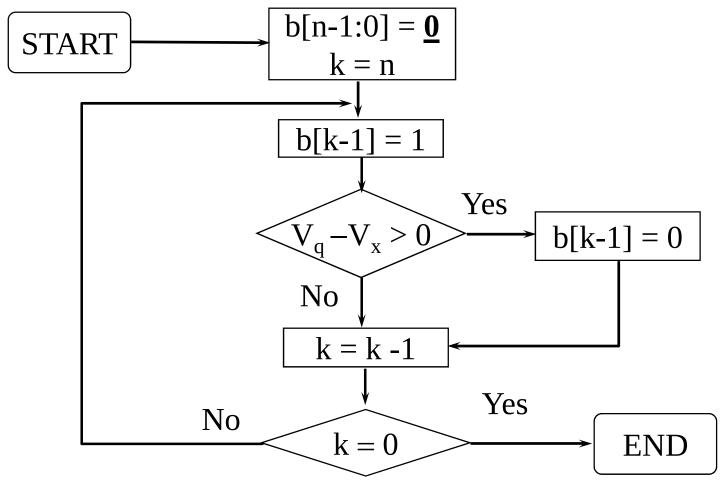
EEL 3744

Successive Approximation



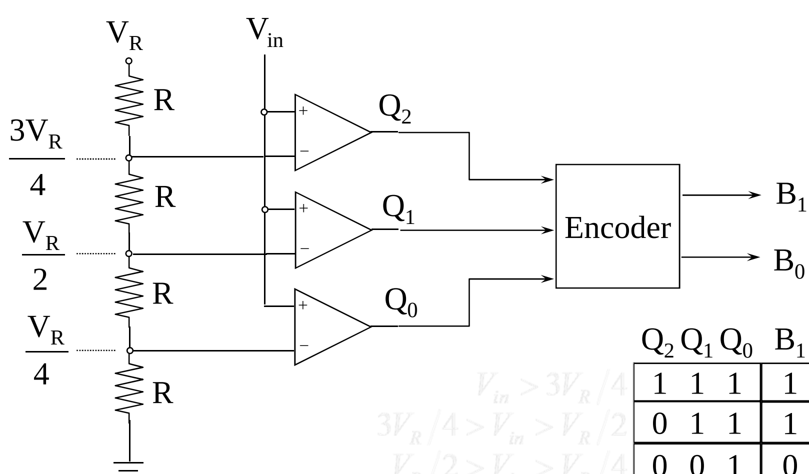
EEL 3744

Successive Approximation Algorithm A.K.A. Binary Search



EEL 3744

Flash Conversion



	Q_2	Q_1	Q_0	B_1	B_0
$V_{in} > 3V_R/4$	1	1	1	1	1
$3V_R/4 > V_{in} > V_R/2$	0	1	1	1	0
$V_R/2 > V_{in} > V_R/4$	0	0	1	0	1
$V_R/4 > V_{in} > 0$	0	0	0	0	0

EEL 3744

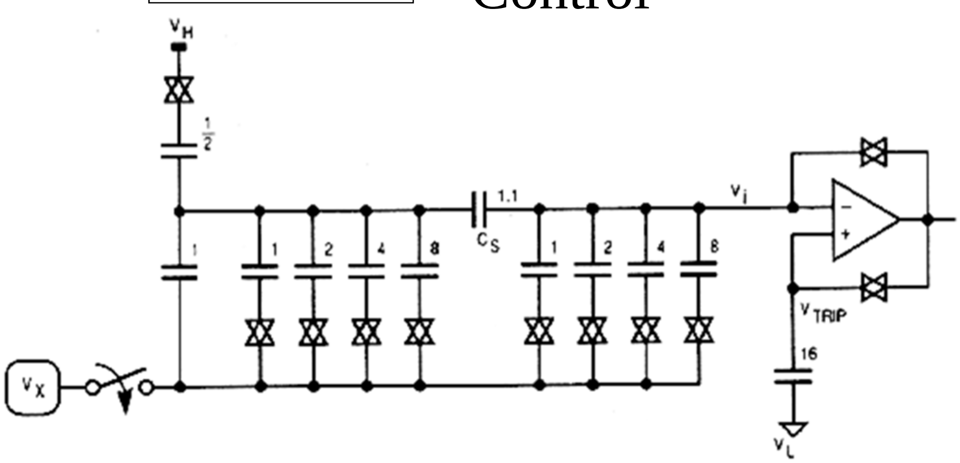
Analog/Digital Conversion

- A/D Conversion Method in the 68HC11
  - >Charge Distribution A/D
    - – See 68HC11 RM Chapter 12

EEL 3744

A/D Successive-Approximation Control

68HC11 RM: Fig.12-3



EEL 3744

The End!