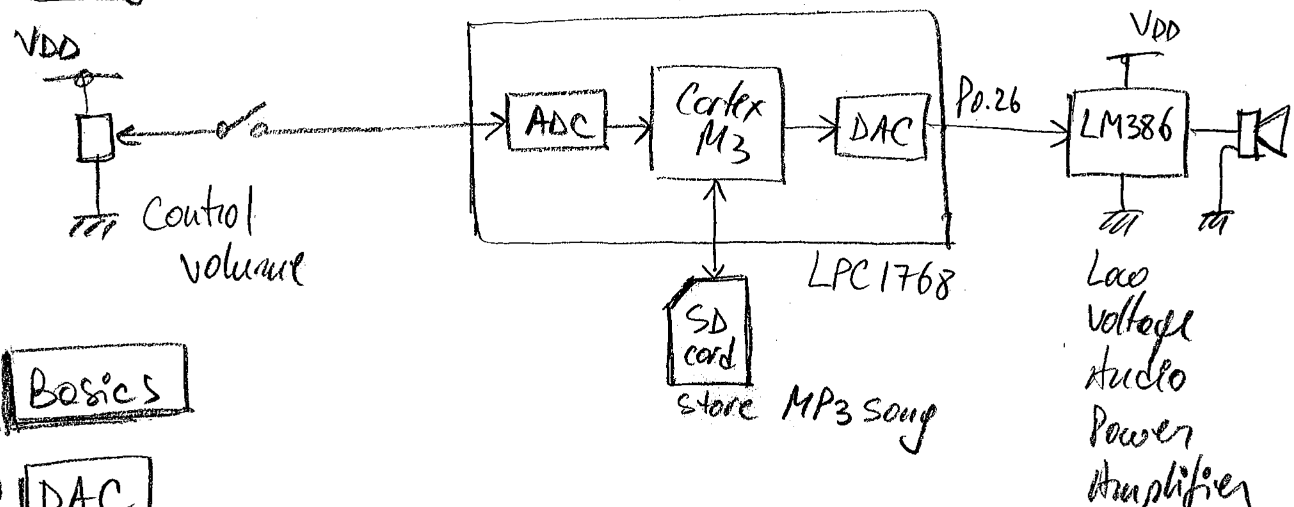
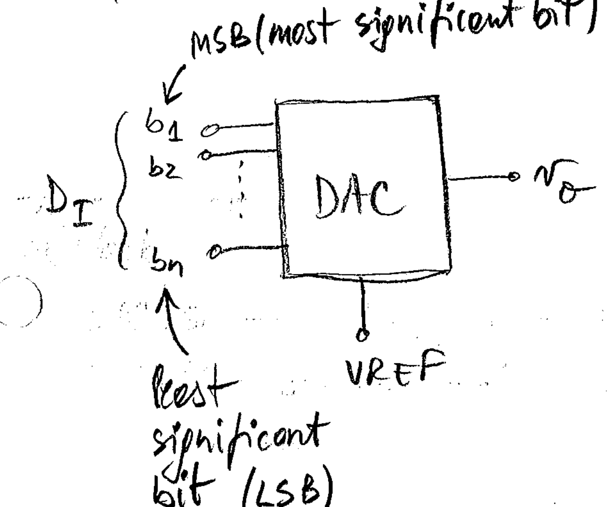
Introduction to (1st talk about IRQ vs. FIQ)

Digital to Analog Converters DAC and Analog to Digital Converters ADC

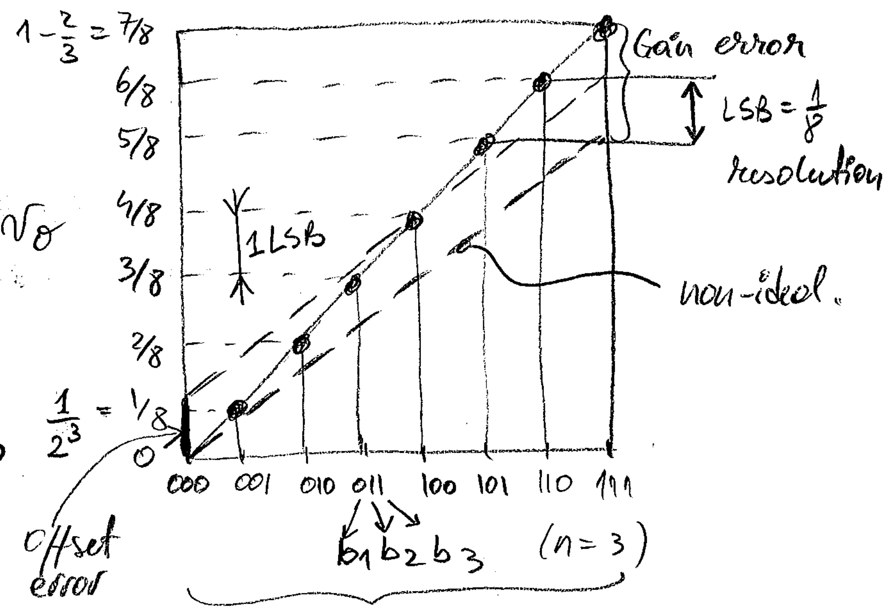


A Basics

1 DAC



Example: $(b_1 b_2 b_3 b_4)_2 = (0111)_2 = 7_{10}$



2^n points => 2^n output values

$$v_0 = kV_{REF} \cdot D_I$$

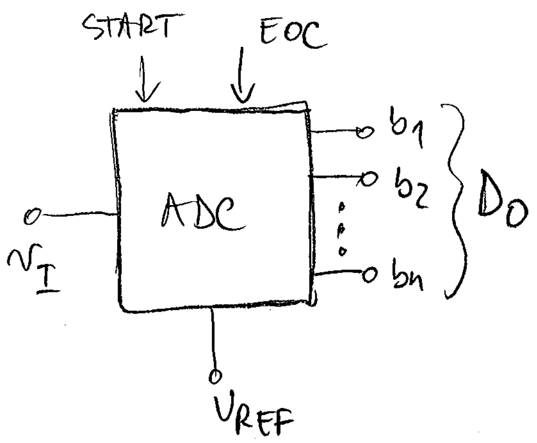
$$= kV_{REF}\left(\underset{MSB}{b_1 \cdot \frac{1}{2}} + b_2 \frac{1}{2^2} + \dots + \underset{LSB}{b_n \frac{1}{2^n}}\right)$$

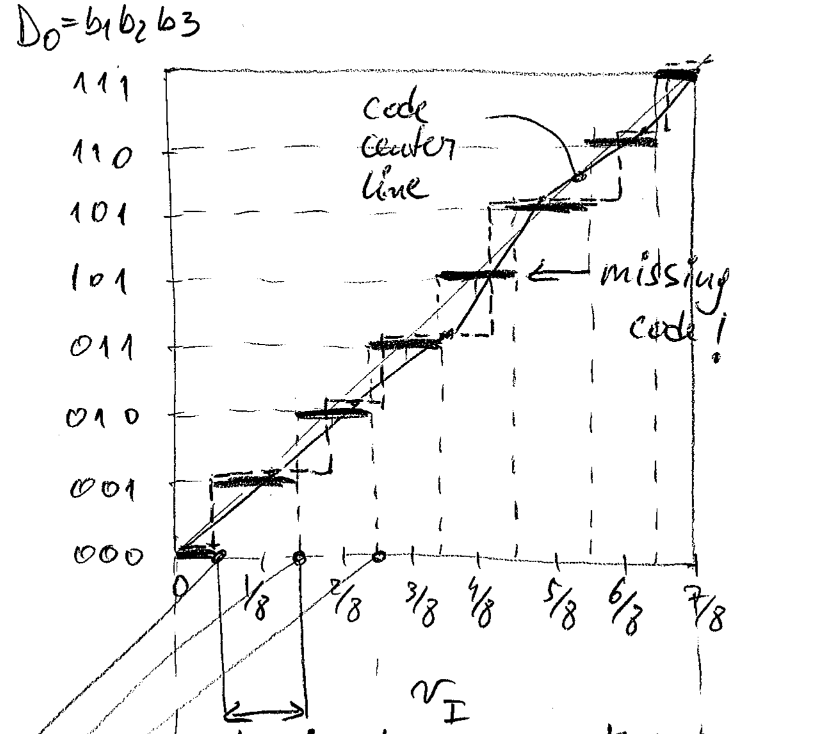
$$= k \cdot V_{REF} \cdot \sum_{i=1}^{n} \frac{b_i}{2^i}$$

Definitions:
- FSR ≡ full scale range $V_{FSR} = kV_{REF}$
- FSU ≡ full scale value $V_{FSU} = (1-2^{-n})V_{FSR}$
- LSB ≡ Resolution $\frac{V_{FSR}}{2^n}$

- This is called a multiplying DAC because v_0 is obtained by multiplying $V_{REF} \times D_I$: $D_I = b_1 \cdot 2^{-1} + b_2 \cdot 2^{-2} + \dots + b_n \cdot 2^{-n}$

② $\boxed{\text{ADC}}$ provides the inverse function of DAC!

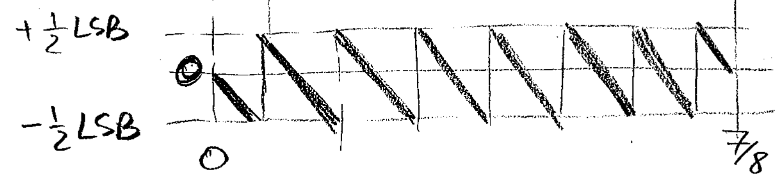




$$D_O = b_1 \frac{1}{2} + b_2 \frac{1}{2^2} + \dots + b_n \frac{1}{2^n}$$

$$= \frac{v_I}{k \cdot V_{REF}} = \frac{v_I}{V_{FSR}}$$

Any input value in this range will be coded as "001"! We have an error ≜ quantization error $\boxed{\text{eq}}$



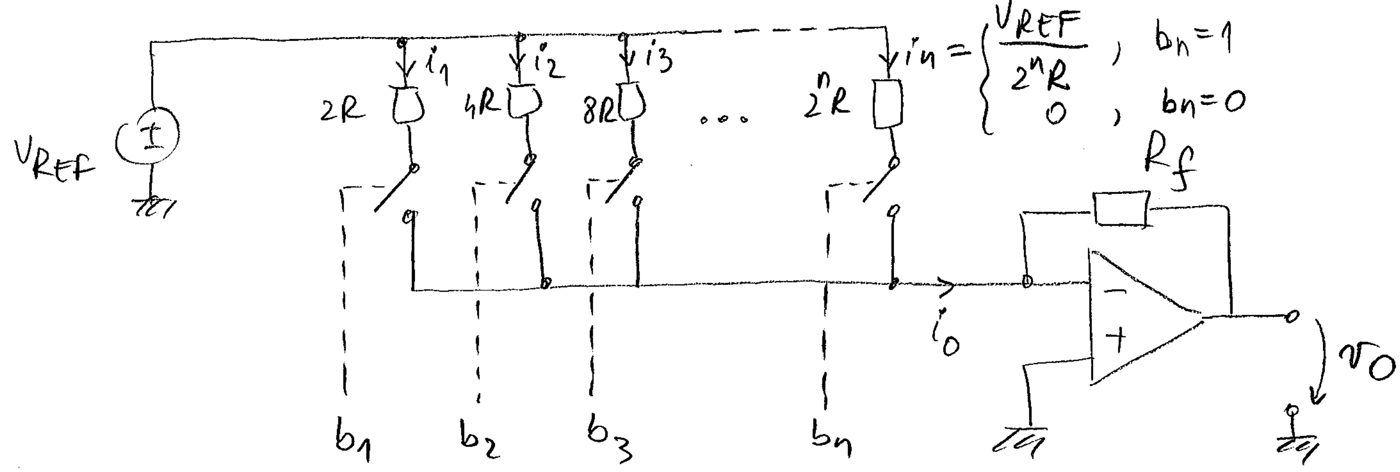
ideally

These transitions take place at <u>odd</u> number of $\frac{1}{2}$LSB; if not, then we have nonlinearities (errors)

Conversion time ≜ time for ADC to complete the transition!

B | DA Techniques

(a) Weighted resistor DAC



$$\frac{v_O}{R_f} = -i_o$$

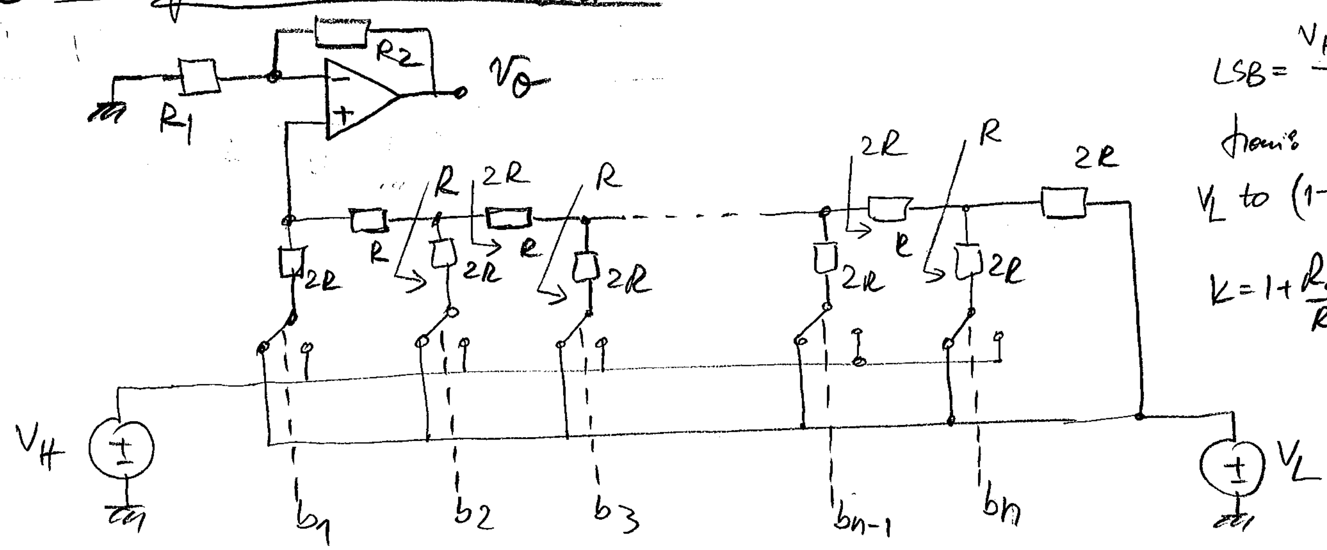
$$v_O = -\frac{R_f}{R} \cdot V_{REF}\left(b_1 \frac{1}{2} + b_2 \frac{1}{4} + b_3 \frac{1}{2^3} + \ldots + b_n \frac{1}{2^n}\right)$$

Drawbacks – nonzero switch resistance
– wide spread of resistor values (exponential increase!)

(b) weighted-capacitor DAC

(c) Potentiometric DAC

(d) Voltage-mode R-2R ladder



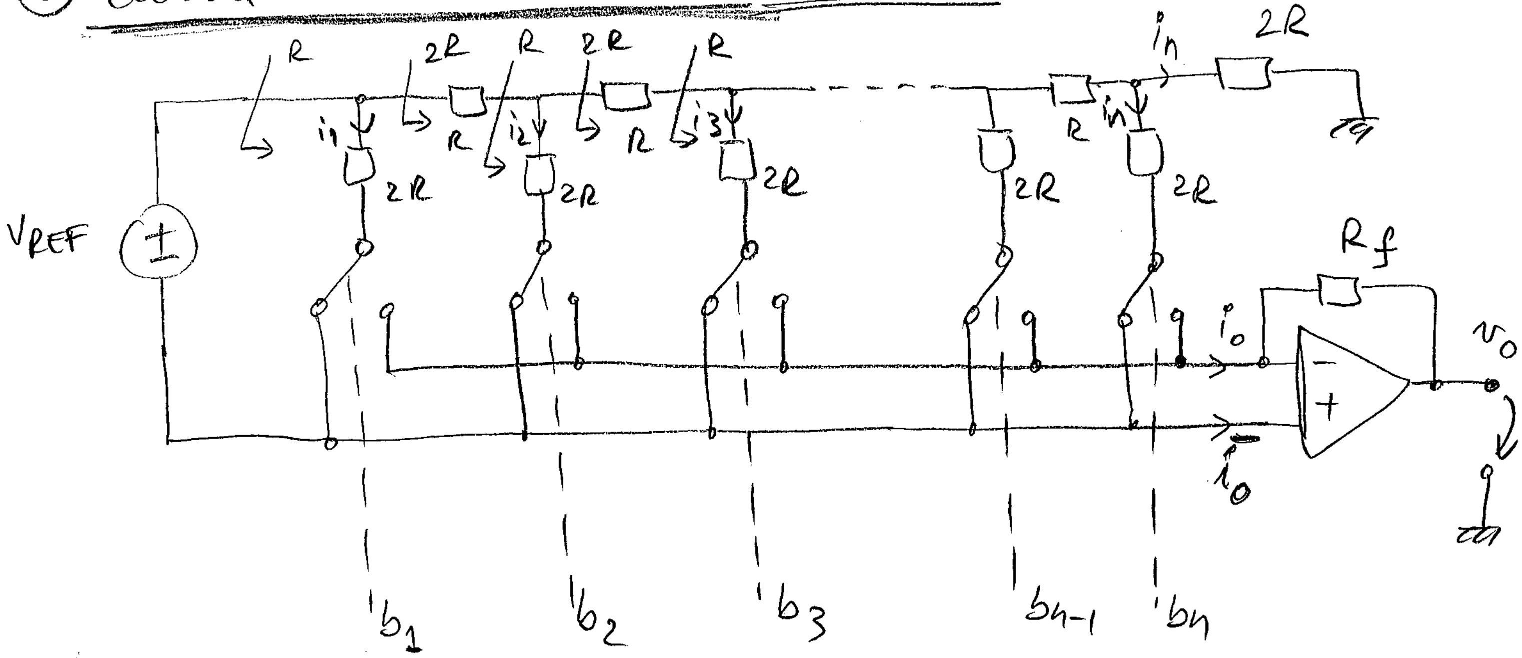
$$LSB = \frac{V_H - V_L}{2^n}$$

from V_L to $\left(1 - \frac{1}{2^n}\right)(V_H - V_L)$

$$K = 1 + \frac{R_2}{R_1}$$

Advantage: we can interpolate between any V_L, V_H!

c) Current-mode R-2R ladder based DA



$$\begin{cases} i_1 = \dfrac{V_{REF}}{2R} = \dfrac{V_{REF}}{R}\cdot\dfrac{1}{2} \\ i_2 = \dfrac{V_{REF}}{4R} = \dfrac{V_{REF}}{R}\cdot\dfrac{1}{2^2} \\ \dots \\ i_n = \dfrac{V_{REF}}{R}\cdot\dfrac{1}{2^n} \end{cases}$$

$$\frac{v_o}{R_f} = -i_o$$

$$v_o = -R_f \cdot i_o$$

$$v_o = -\frac{R_f}{R} V_{REF}\cdot\left(b_1\frac{1}{2} + b_2\frac{1}{2^2} + \dots + b_n\frac{1}{2^n}\right)$$

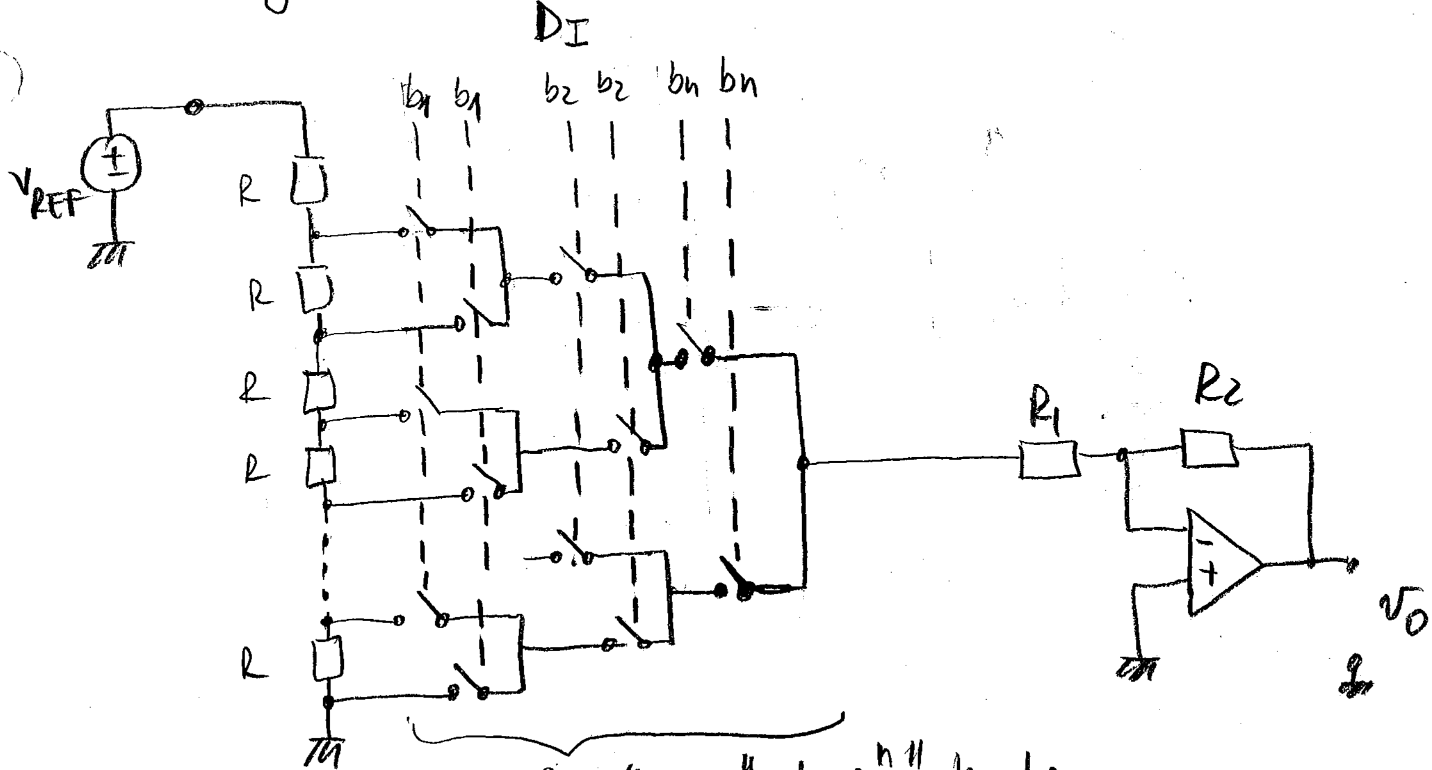
$$\triangleq k = -\frac{R_f}{R}$$

$$\boxed{v_o = k V_{REF}\left(b_1\frac{1}{2} + b_2\frac{1}{2^2} + \dots + b_n\frac{1}{2^n}\right)}$$

OBS: - Note that $i_o + \bar{i}_o = (1-2^{-n})\cdot\frac{V_{REF}}{R}$ regardless of $D_I = b_1 b_2 b_3 \dots b_n$!

- $\bar{i}_o$ is said to be complementary to i_o

(f) R-String architecture (used inside LPC1768)



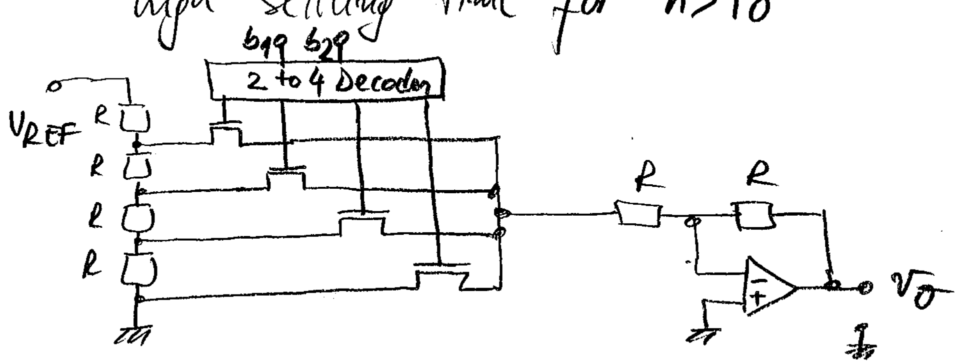
≡ Basically a "n to 2^n" decoder.

Advantages: – simple (needs only identical resistors)
- fast for < 8–10 bits
- monotonic (inherently)
- compatible w/ purely digital technologies
- low power

Disadvantages:

- 2^n resistors and 2^n switches for n bits
  ⇒ high element count, high area!
- high settling time for $n > 10$

Example:



C AD techniques

a) DAC based AD conversion:
"Successive-approximation converters" ← Uses SAR (successive approximation register)

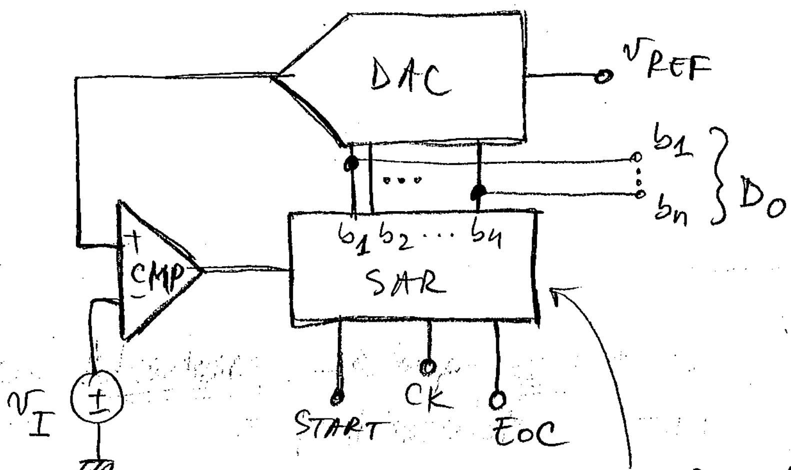
b) charge-redistribution converters

c) flash converters

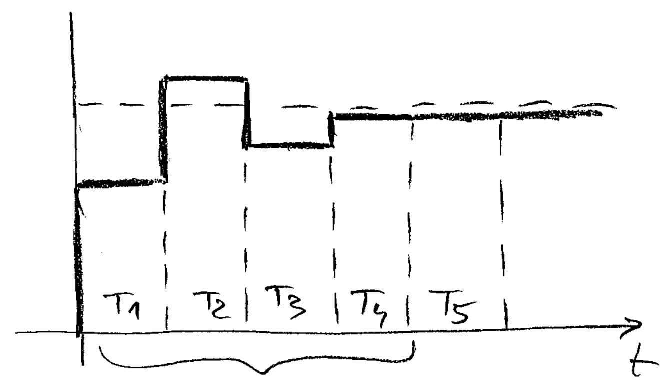
d) subranging converters

e) integrating-type converters.

a) DAC based A-D Conversion (used by LPC 1768)



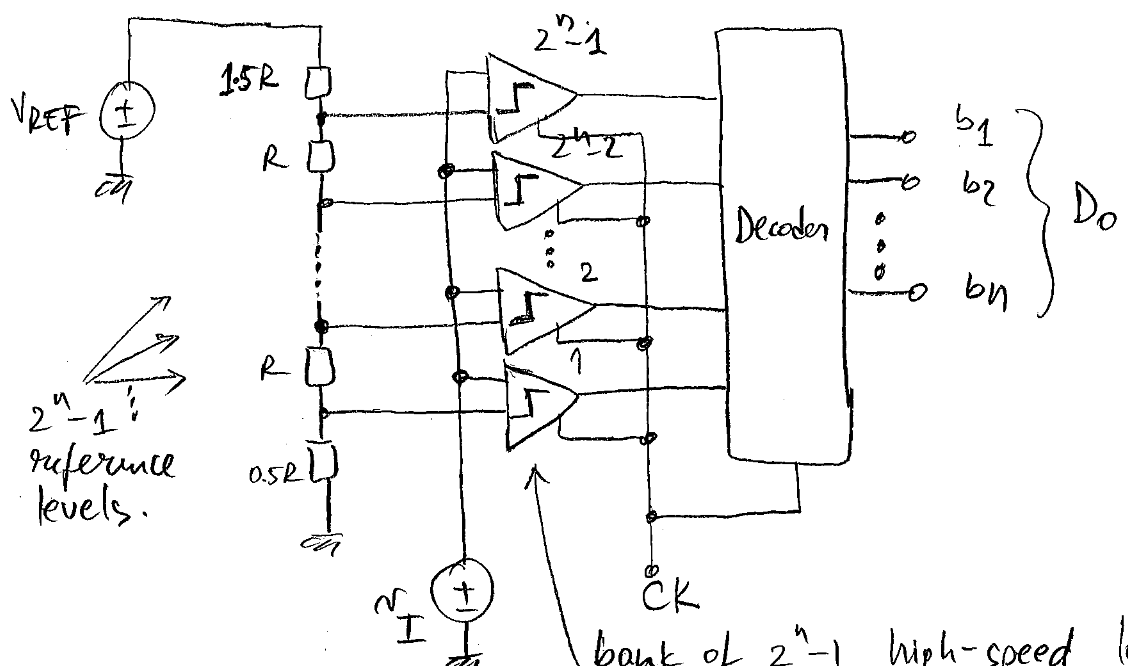
Example:



4 clock cycles.

SAR adjusts DAC's input code until DAC's output comes within ½ LSB of input voltage v_I !

c) Flash converters



bank of 2^n-1 high-speed latched comparators!

LPC 1768

DAC - 10 bit
- resistor string architecture
- maximum update rate 1 MHz

ADC: - 12-bit successive approximation.
- conversion rate 200 kHz

$\boxed{\text{IRQ vs. FIQ}}$ fast interrupt - (available in ARM 7/9/10/11)
- not available in cortex-M3
- however, interrupt priority handling and nested interrupt support are included in the "interrupt support"

- FIQ - fast interrupt, higher priority (than IRQ) which is prioritized by disabling IRQ during request servicing; patented by ARM.
- IRQ - normal priority