EEE3096S: Embedded Systems II

LECTURE 16: THE ADC (PART 1)

Presented by:

STANLEY MBEWE

Electrical Engineering
University of Cape Town

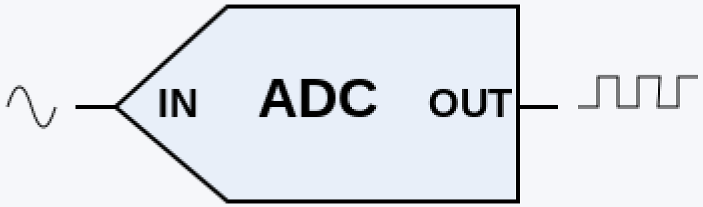
OUTLINE OF LECTURE

- ADC – basic conceptual model
- ADCs Design Flavours*
  - Flash ADC
  - Multi-stage Flash ADC
  - Successive Approximation ADC (SA-ADC)
  - Pipelined Flash ADC

* Note that there are many other design of ADCs, these slides review some of the most common approaches.

This presentation contains some slides adapted from course textbook "Embedded System Design Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things" by Peter Marwedel

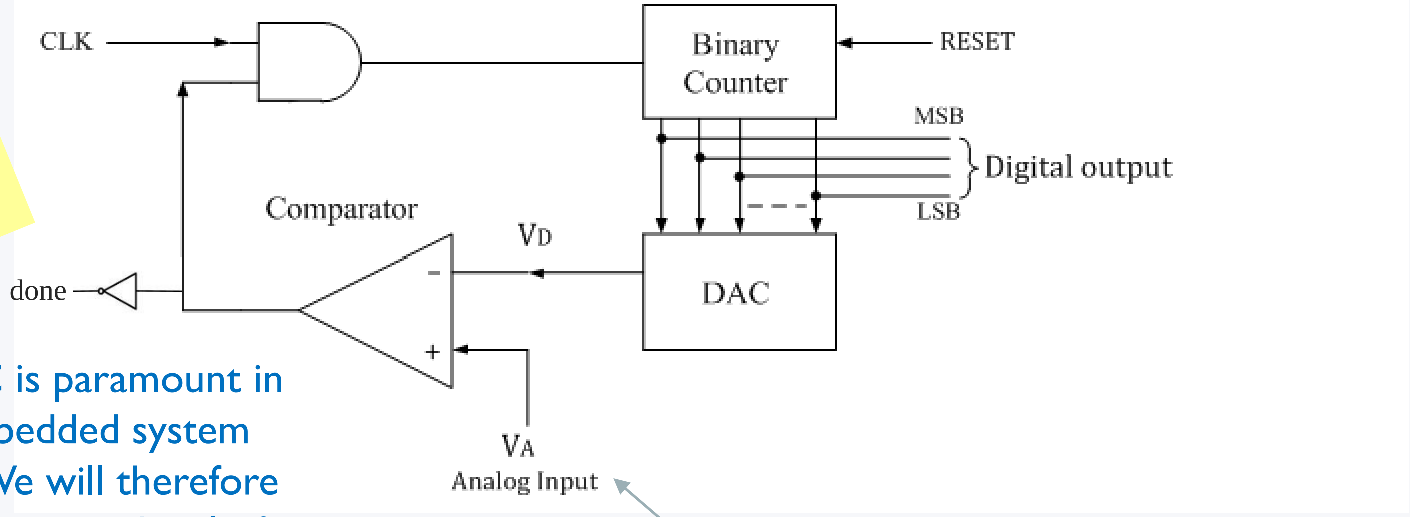
ADCs



Analogue to Digital Convertors (ADCs)

CONCEPTUAL ADC OPERATION

Main point: a common approach is the use of a DAC within an ADC. (we will present DACs next week)



The ADC is paramount in many embedded system designs. We will therefore delve into some detail of ADC issues.

Sampled analogue signal

Notes about this conceptual representation:

- *While this conceptual ADC could be implemented, it's not such a practical architecture! It is very inefficient and only really useful to illustrate the operation of an ADC.*
- *In practice, a sample-and-hold (SAH) circuit grabs the ADC input, a time-varying analogue voltage x(t), and prevents it from changing during conversion.*
- *The counter stops counting when its output, which is converted into a voltage by the DAC (digital-to-analogue converter), equals x(t), indicated by comparator. The counter output is then the ADC output code corresponding to x(t).*

[2]

CONCEPTUAL ADC OPERATION

- Example DAC output voltages from the changing binary counter values (i.e. the DAC’s transfer function) shown in the table below:

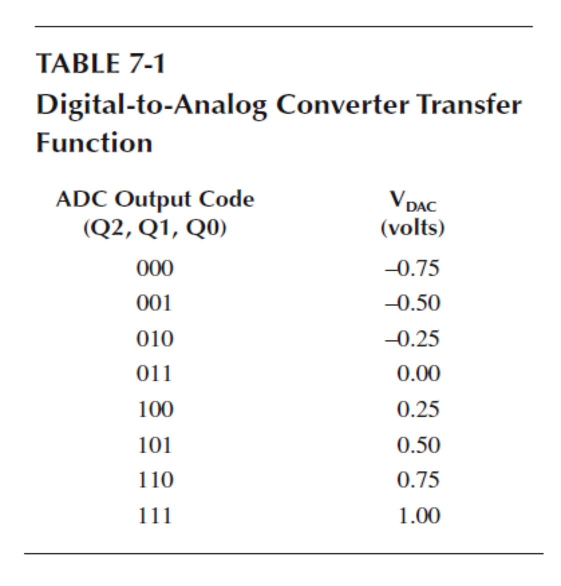


TABLE 7-1
Digital-to-Analog Converter Transfer Function

ADC Output Code (Q2, Q1, Q0)	V_{DAC} (volts)
000	–0.75
001	–0.50
010	–0.25
011	0.00
100	0.25
101	0.50
110	0.75
111	1.00

Basically you need to remember:

digital Output Codes corresponds to a input voltage and there will be metrics that described how these relate…

Notes about this conceptual representation:

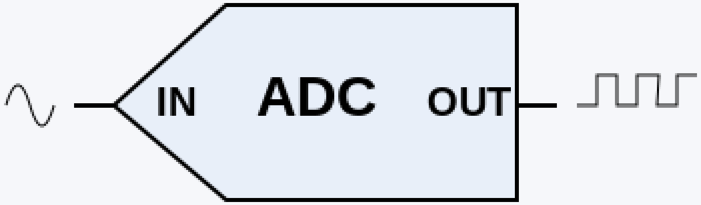
In this conceptual design, the least significant bit (LSB), Q0, is the step size of 0.25 volts. This can also be quite an artificial assumption for a real system where the step size may fluctuate due to the physical properties of the system and environment. [2]

ADCs Flavours

What flavour is your ADC?

You want Raspberry Pi flavour? Sorry, isn't a flavour; try another.

The Raspberry Pi doesn’t have a built-in ADC alas.

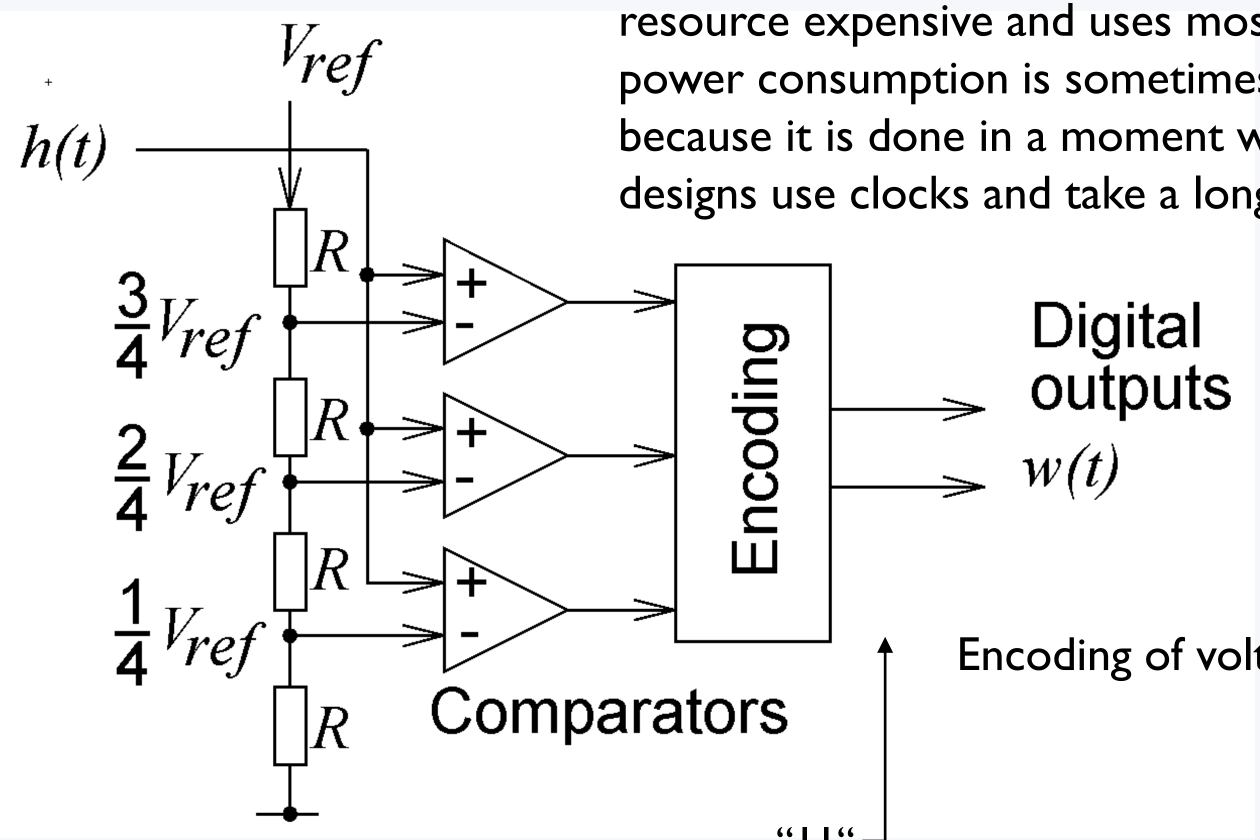


IN ADC OUT

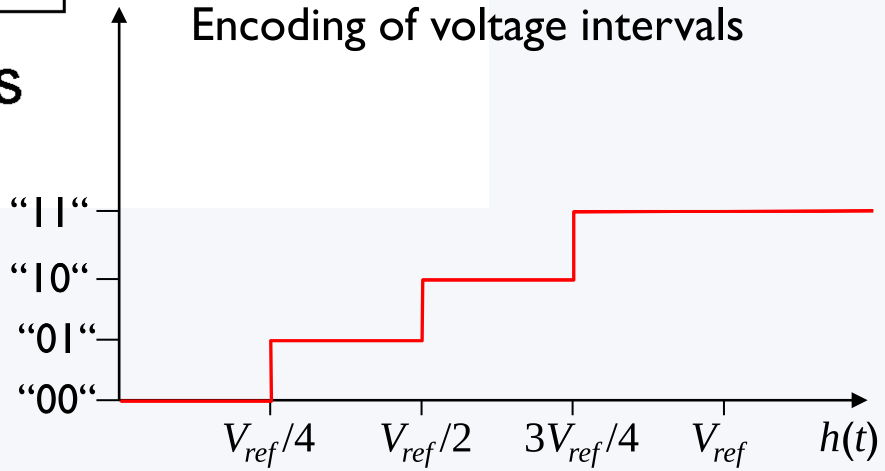
Analogue to Digital Convertors (ADCs)

FLASH ADC CONVERTER

The Flash ADC is essentially the fastest but most resource expensive and uses most power. But the power consumption is sometimes not trivial because it is done in a moment whereas the other designs use clocks and take a longer time.



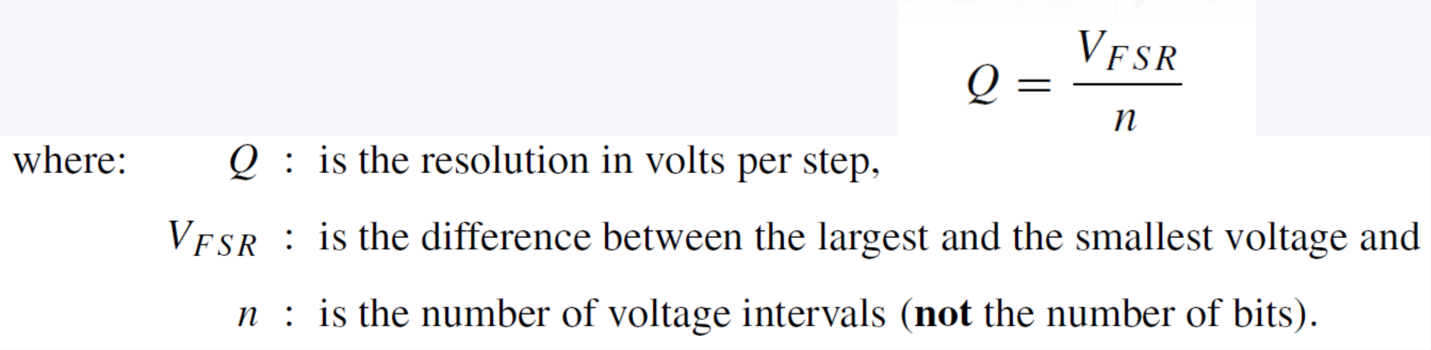
Encoding of voltage intervals



No decoding of $h(t) > V_{ref}$

RESOLUTION OF FLASH A/D-CONVERTER

Number of bits produced. But can also be measured in volts by the input voltage that causes the output to be increment by 1.



$$Q = \frac{V_{FSR}}{n}$$

where: Q : is the resolution in volts per step,

V_{FSR} : is the difference between the largest and the smallest voltage and

n : is the number of voltage intervals (**not** the number of bits).

We need $n - 1$ comparators to distinguish n values. For high resolutions, we need a lot of comparators which may not be physically possible in ICs, hence these must be built differently.

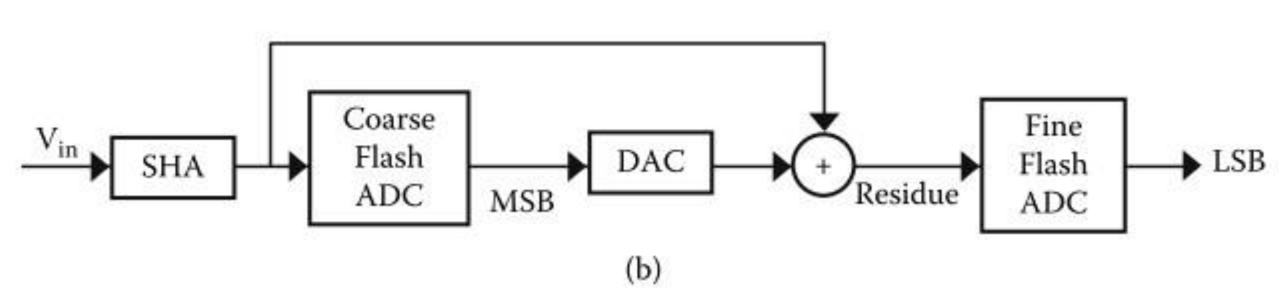
Brief Properties

- Parallel comparison with reference voltage
- Speed: O(1)
- Hardware complexity: O(n)

Applications: e.g. in video processing

MULTI-STAGE FLASH (NB: NOT PIPELINED)

- A multistage flash ADC uses a number of smaller flash ADCs to decide on a sample, but this isn’t quite as fast as the pure flash



Example of 2-stage residual FLASH

Speed is depended on:

Course Flash + DAC + Residual + Fine Flash

which is about 1/3 the speed for one single big flash

But on the positive side:

single flash = 2^n comparators e.g. 8-bit → 256 comparators

s-stage flash = 2 * 2^(n/2) → e.g. 8-bit → 2 * 2^4 = 2*16 = 32 comparators

So for 8x the resources you get about 3x the speed

PIPELINED FLASH

Generally, this is the best bang for your buck… but

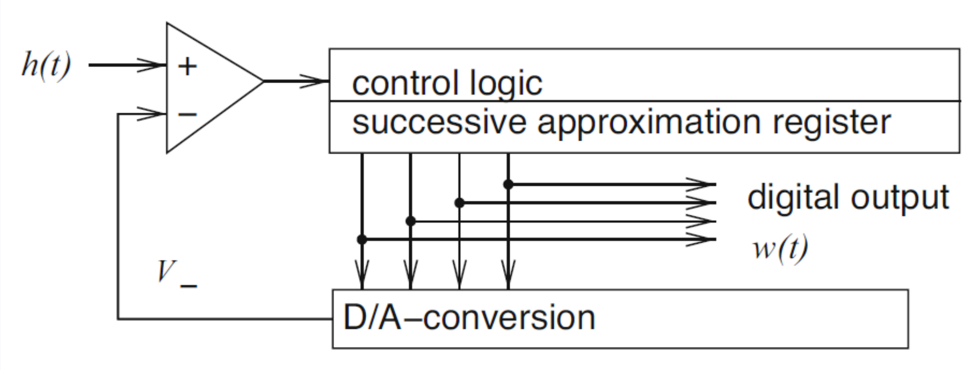
I'm the FLASH ADC
I might be… a little heavy,
But I'm FAST!

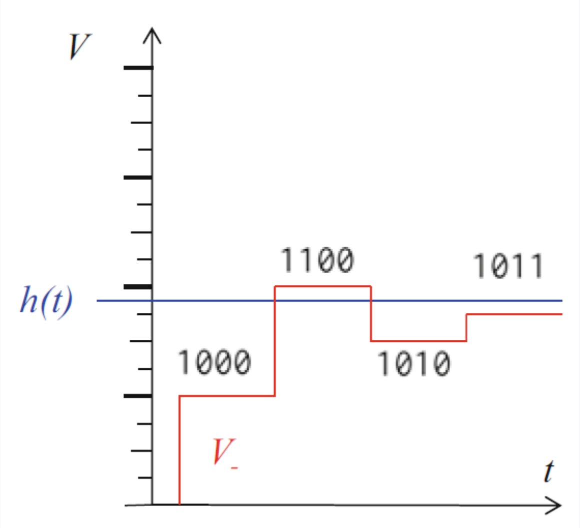
We're the Pipeline FLASH
We're almost as FAST
But we're a little cheeky!

But to understand the Pipeline Flash ADC
you need to first understand the SA-ADC
(i.e. the Successive Approximation ADC) which is up next…

The Pipeline Flash is essentially a combination of the two

SUCCESSIVE APPROXIMATION ADC (SA-ADC)





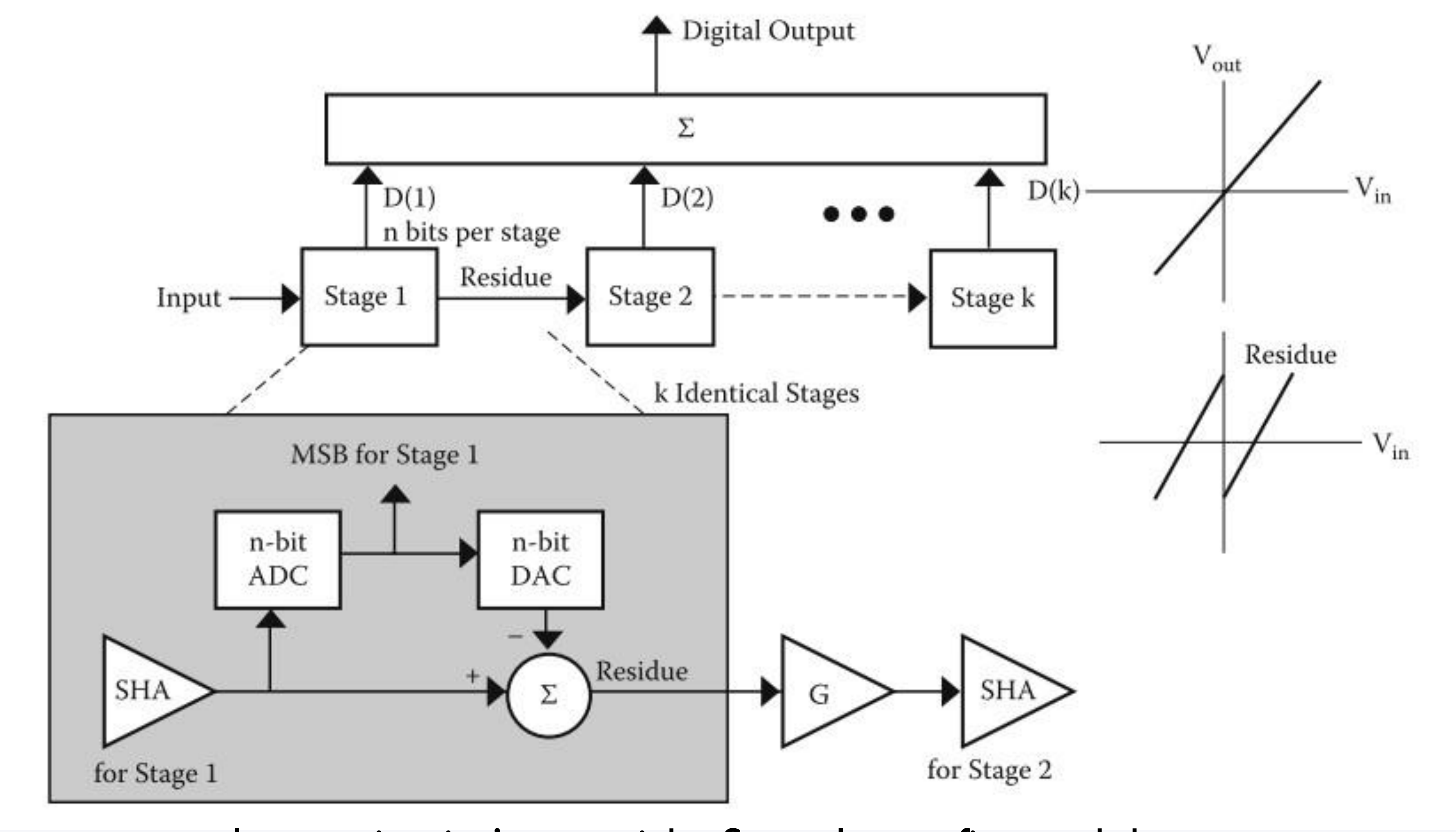
Key idea: binary search…

```
Set MSB='1' (all other ='0')
if too large:
    reset MSB, set MSB-1='1'
    if too large:
        reset MSB-1, set MSB-2='1'
        etc..
else if too small:
    set MSB-1='1' …
```

Brief Properties

- Speed: O(log2(n))
- Hardware complexity: O(log2(n))
  - with n= # of distinguished voltage levels;
- slow, but high precision possible.

PIPELINED FLASH



The conceptual operation isn't too tricky. Stage 1 runs first and does a coarse estimate. Then stage 2 runs and does a fine estimate … but simultaneously stage 1 start a course estimate on a new sample. The key aspect here is the SAH at each stage, essentially a smaller and smaller residual moves down the pipeline until the final LSB is decided. I think it's a really cleaver hybrid digital+analogue design!!

QUICK UNDERSTANDING CHECK

Q1. Does a SA-ADC usually use more comparators than a standard (i.e. 'fat') Flash?

Q2. Will a pipeline Flash probably be quicker than a SA-ADC? (explain your answer)

Answers…

QUICK UNDERSTANDING CHECK

Q1. Does a SA-ADC usually use more comparators than a standard (i.e. 'fat') Flash?

No. A FAT flash needs n-1 comparators, whereas a SA-ADC, a bit slower, needs log2(n), i.e. in it pipeline of log(n) hot/cold comparators.

Q2. Will a pipeline Flash probably be quicker than a SA-ADC? (explain your answer)

Yes, it does the whole kit and caboodle in one go, about log2(n) times faster than the SA-ADC. The voltage input to the FAT flash is tested by all n-1 comparators in one go. Whereas for the SA-ADC the voltage moves down the line being tested for too high / too low in each stage until it gets to the last stage and final bit decision.