



COMP6733

IoT Design Studio

Embedded Signal Processing

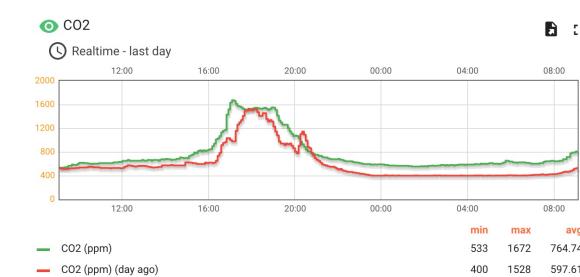
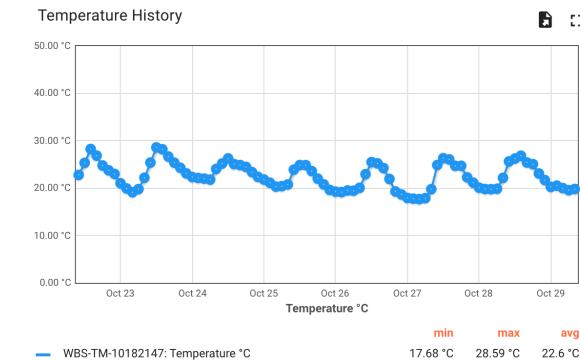
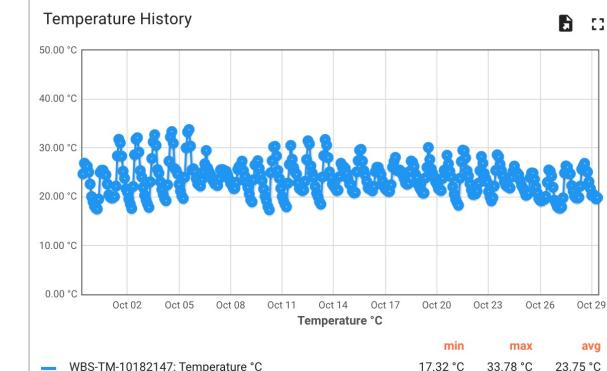
Signal

- variation of a physical phenomenon/quantity with respect to one or more independent variables
- pattern of variation of some form
- variable carrying information

A (discrete) signal is a sequence of numbers representing the value of a physical variable over time or another physical variable.

examples:

- electric current, voltage
- electromagnetic radiation, radio wave, light
- acoustic pressure wave, sound, ultrasound



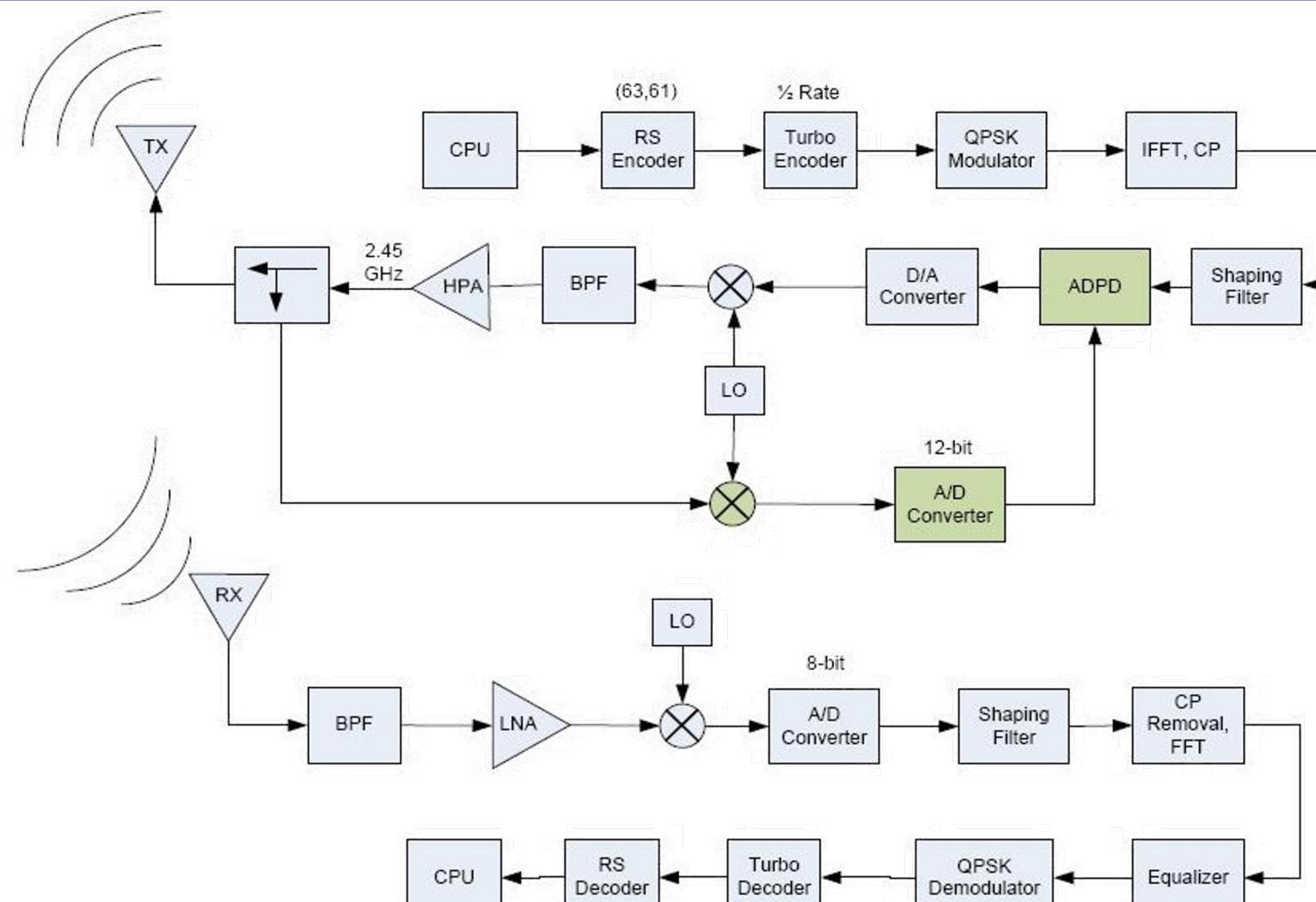
Signal Processing

manipulation of a signal to **manage** the information it contains

common tasks:

- **acquisition**: sampling, quantising
- **enhancement**: noise reduction, echo cancellation, filtering, ...
- **parameter estimation**: feature extraction, event detection, system identification, ...
- **(de)compression**: usually for multimedia
- **(de)coding**: error-correction coding, channel coding, ...
- **encryption/decryption**: scrambling, hashing, ...
- **(de)modulation**
- **(de)multiplexing**
- **synchronisation**

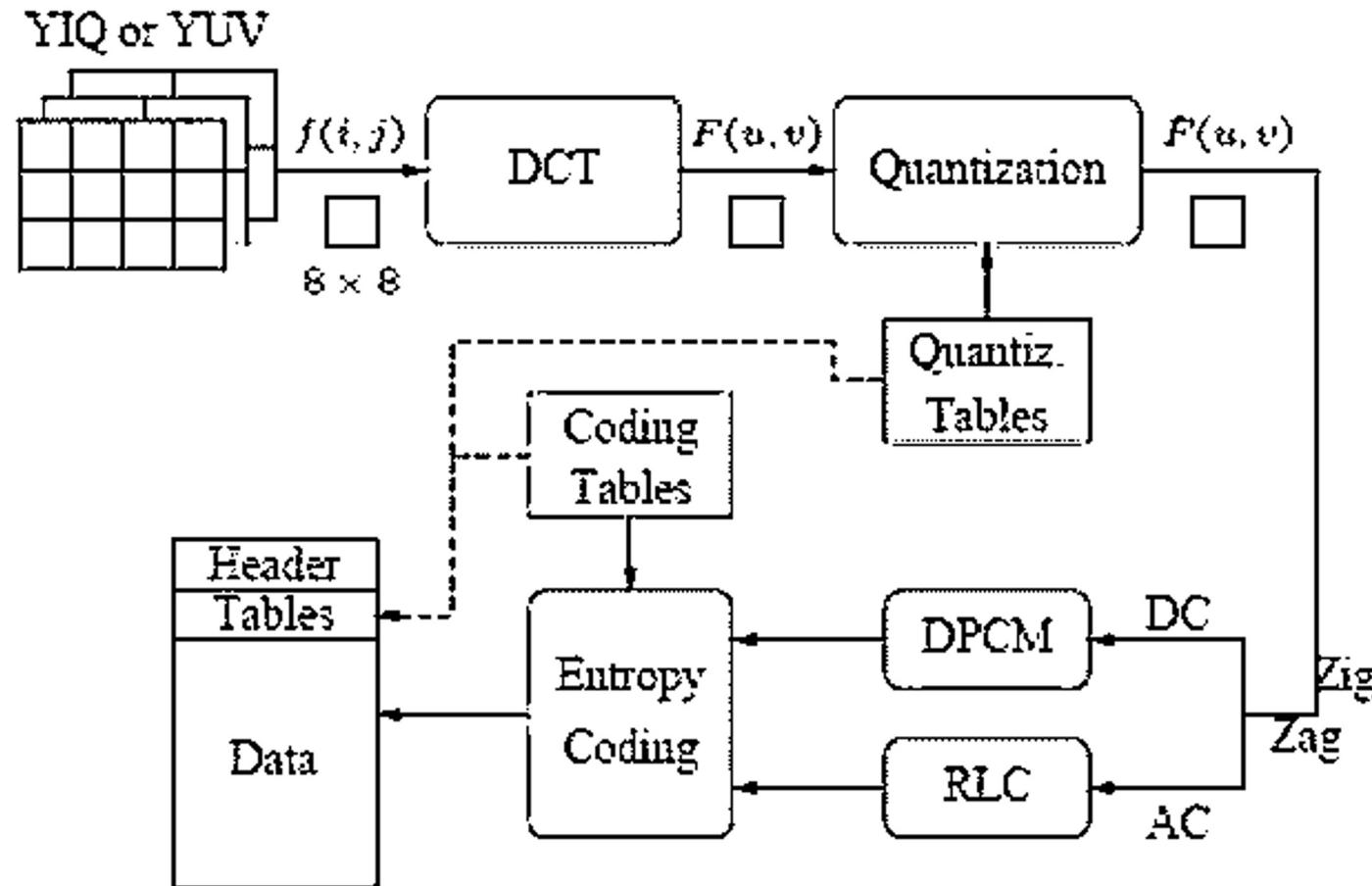
Example Communication System



communication link between astronaut and lunar lander

<http://www.propagation.gatech.edu/ECE6390/project/Fall2009/SELENE/SELENE/communication.html>

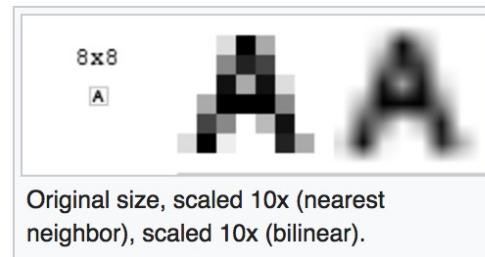
JPEG Image Compressor



Discrete Cosine Transform (DCT)

$$F[m, n] = \frac{2}{\sqrt{MN}} C(m) C(n) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f[x, y] \cos\left(\frac{(2x+1)m\pi}{2M}\right) \cos\left(\frac{(2y+1)n\pi}{2N}\right)$$

Example



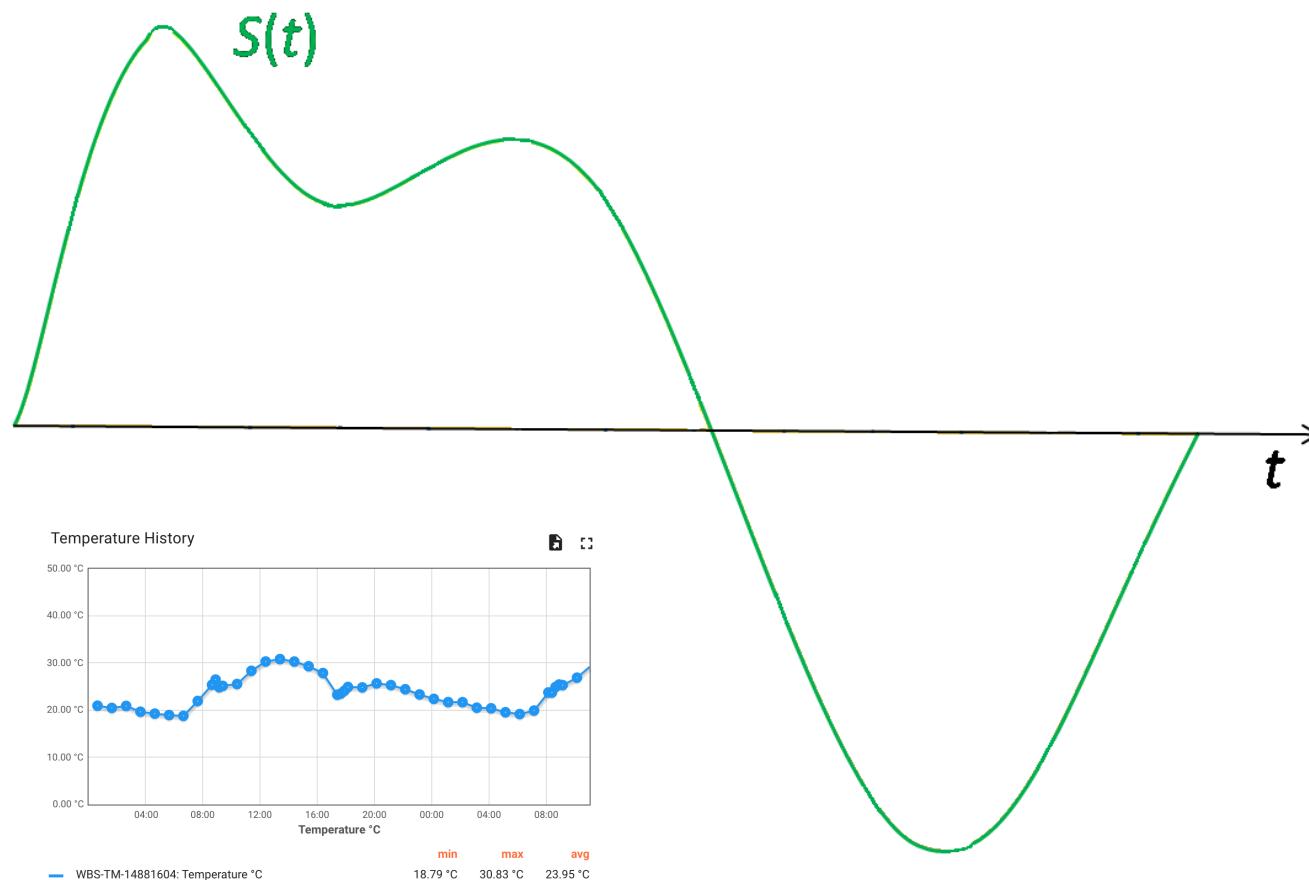
6.1917	-0.3411	1.2418	0.1492	0.1583	0.2742	-0.0724	0.0561
0.2205	0.0214	0.4503	0.3947	-0.7846	-0.4391	0.1001	-0.2554
1.0423	0.2214	-1.0017	-0.2720	0.0789	-0.1952	0.2801	0.4713
-0.2340	-0.0392	-0.2617	-0.2866	0.6351	0.3501	-0.1433	0.3550
0.2750	0.0226	0.1229	0.2183	-0.2583	-0.0742	-0.2042	-0.5906
0.0653	0.0428	-0.4721	-0.2905	0.4745	0.2875	-0.0284	-0.1311
0.3169	0.0541	-0.1033	-0.0225	-0.0056	0.1017	-0.1650	-0.1500
-0.2970	-0.0627	0.1960	0.0644	-0.1136	-0.1031	0.1887	0.1444

Basis functions of the discrete cosine transformation with corresponding coefficients (specific for our image).

$$\text{DCT of the image} = \begin{bmatrix} 6.1917 & -0.3411 & 1.2418 & 0.1492 & 0.1583 & 0.2742 & -0.0724 & 0.0561 \\ 0.2205 & 0.0214 & 0.4503 & 0.3947 & -0.7846 & -0.4391 & 0.1001 & -0.2554 \\ 1.0423 & 0.2214 & -1.0017 & -0.2720 & 0.0789 & -0.1952 & 0.2801 & 0.4713 \\ -0.2340 & -0.0392 & -0.2617 & -0.2866 & 0.6351 & 0.3501 & -0.1433 & 0.3550 \\ 0.2750 & 0.0226 & 0.1229 & 0.2183 & -0.2583 & -0.0742 & -0.2042 & -0.5906 \\ 0.0653 & 0.0428 & -0.4721 & -0.2905 & 0.4745 & 0.2875 & -0.0284 & -0.1311 \\ 0.3169 & 0.0541 & -0.1033 & -0.0225 & -0.0056 & 0.1017 & -0.1650 & -0.1500 \\ -0.2970 & -0.0627 & 0.1960 & 0.0644 & -0.1136 & -0.1031 & 0.1887 & 0.1444 \end{bmatrix}$$

Signal Acquisition

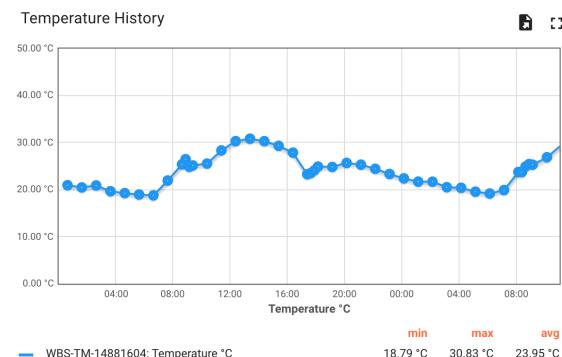
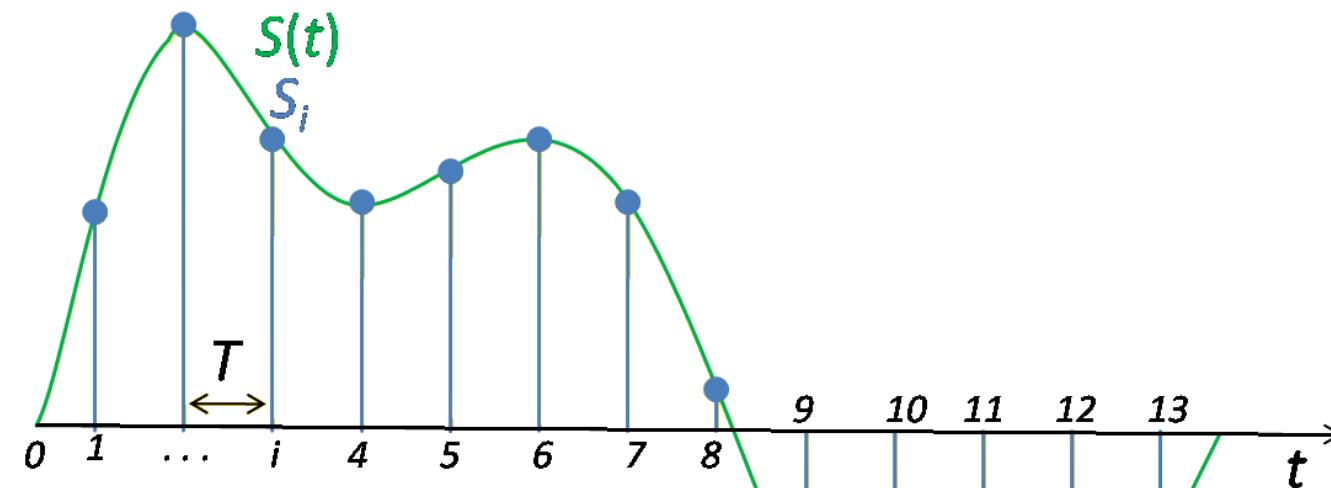
real-world signals are often **analogue**, i.e., have continuous values over a continuous domain



Sampling

extracting samples from a continuous signal

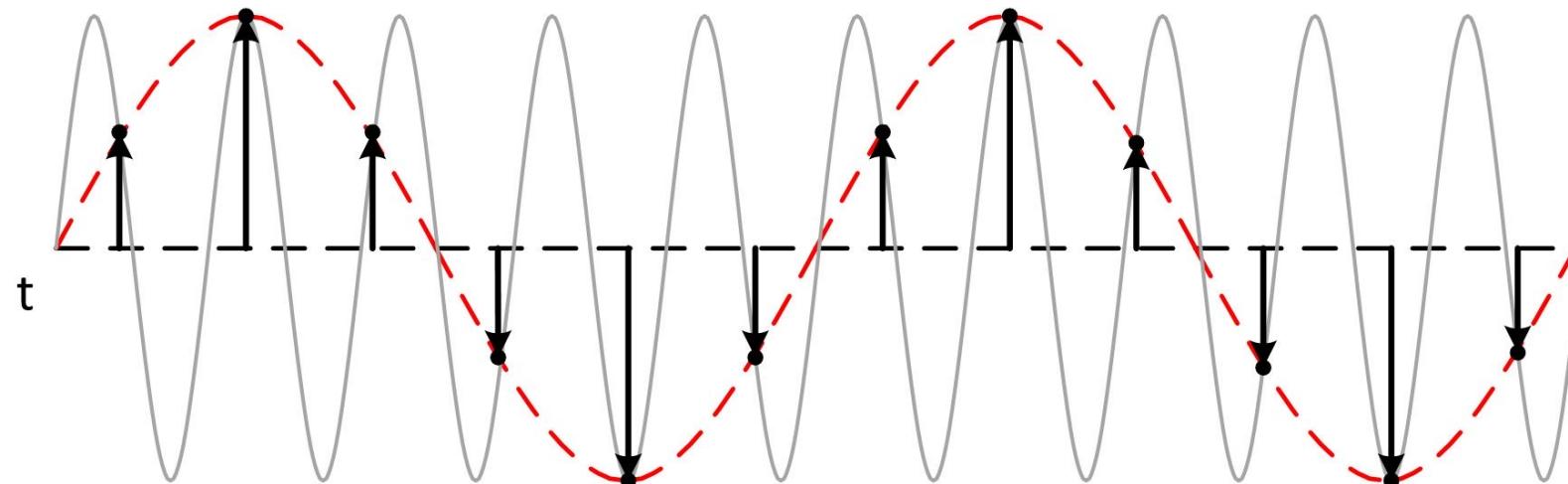
continuous-time signal \rightarrow **discrete-time** signal



Sampling

ideal sampler: produces samples equivalent to the instantaneous value of the continuous signal at the desired points

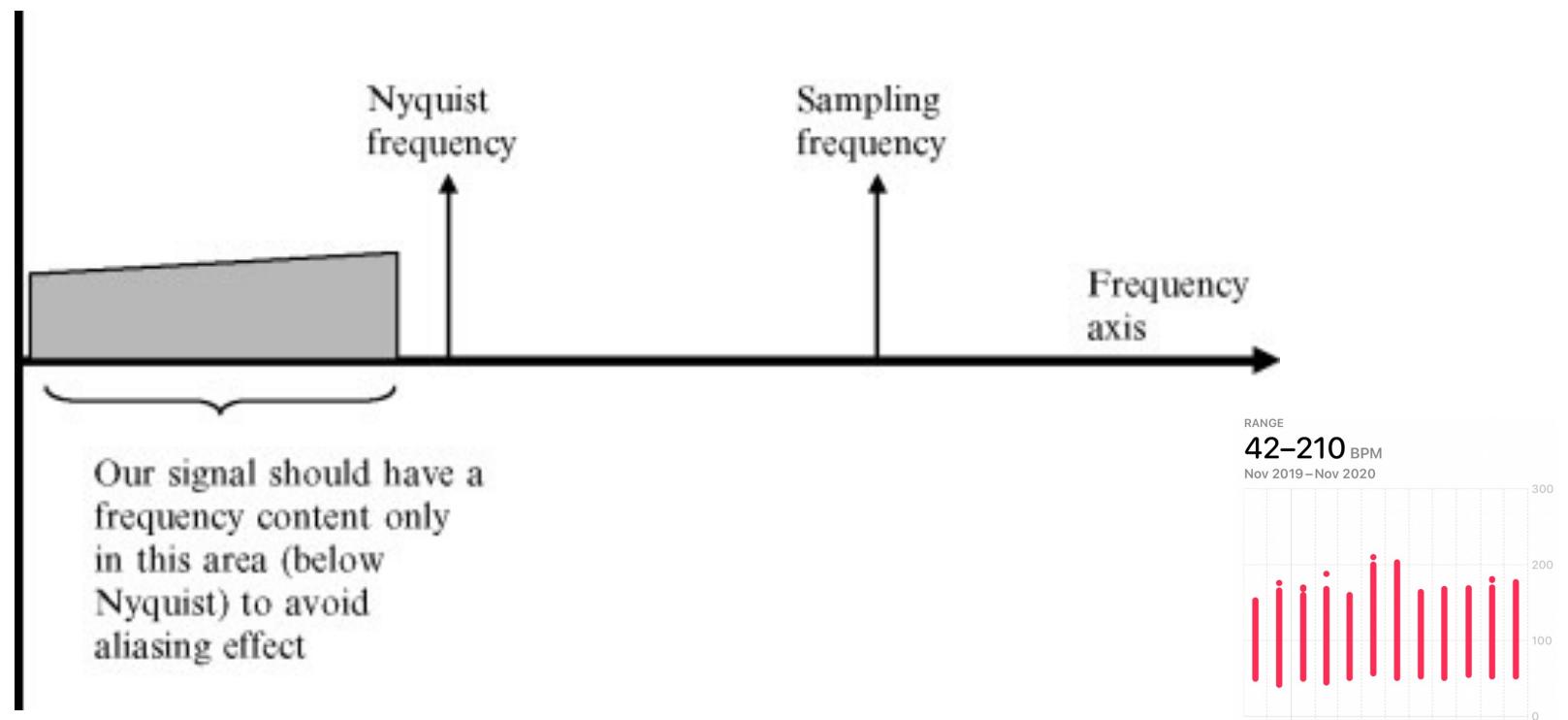
aliasing: causes different signals to become indistinguishable (or *aliases* of one another) when sampled



Sampling

Nyquist–Shannon theorem:

If a function $x(t)$ contains no frequencies higher than B Hertz, it is completely determined by giving its ordinates at a series of points spaced $T = 1/(2B)$ seconds apart.



Sampling

common audio sample rates:

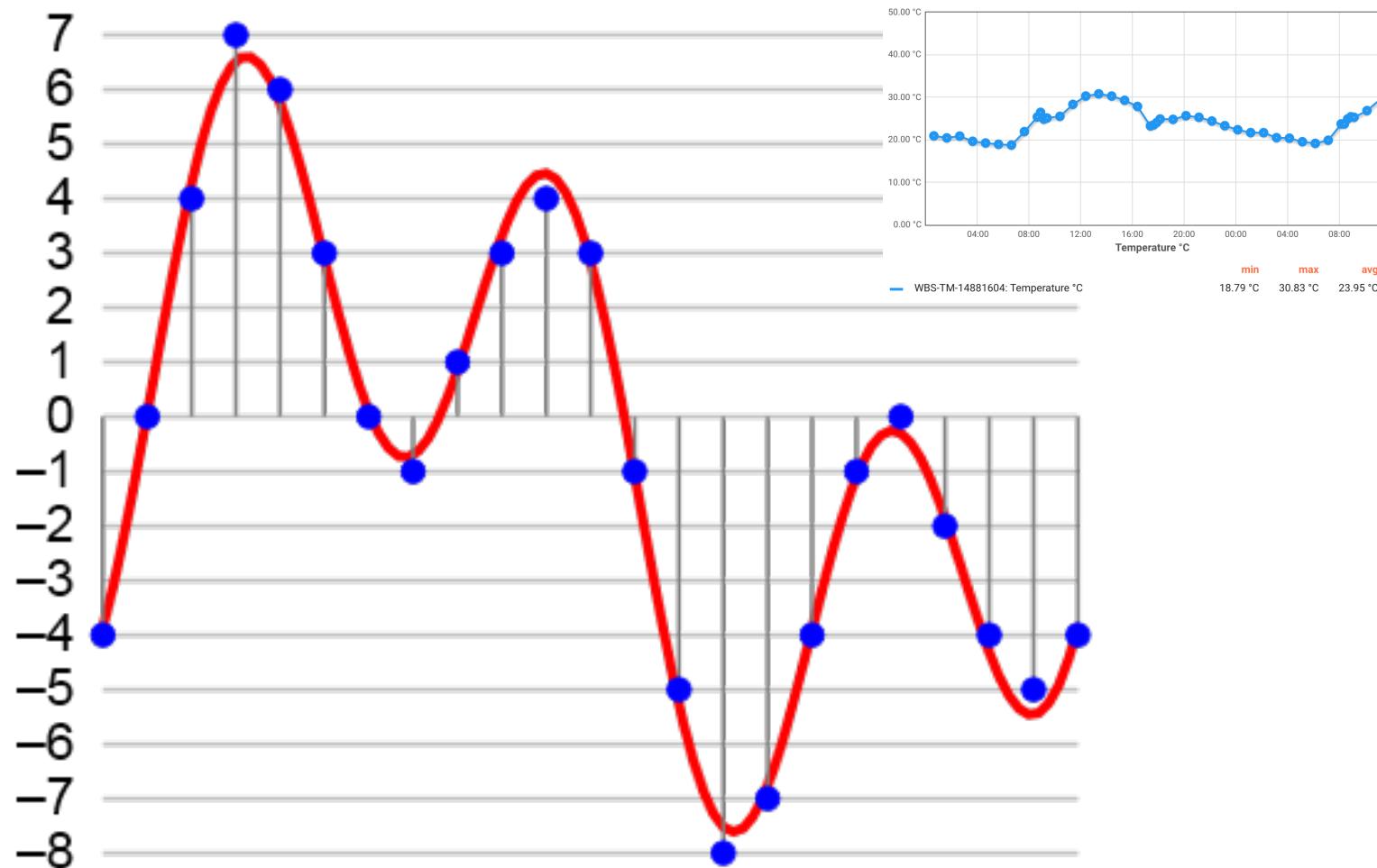
sampling rate (sample/s)	application
8,000	telephone
16,000	voice over IP
44,100	audio CD
96,000	Blu-ray audio track

common video sample rates:

sampling rate (pixel/ s)	application
$720 \times 576 \times 25$	DVD-video, DVB-T (digital TV)
$1920 \times 1080 \times 25$	Blu-ray video, HD DVB-T
$3840 \times 2160 \times 50$	UHD Blu-ray video

Quantisation

continuous-value (analogue) signal → discrete-value (**digital**) signal



Quantisation

- Quantisation error (noise) for a fixe-length code using N -bit:
$$SQNR (dB) = 20 \log_{10} 2^N = 6.0206 \times N$$
- For each extra bit, approximately 6 dB improvement in SQNR

N	$SQNR (dB)$	possible integer values	base 10 signed range
4	24.08	16	-8 to +7
8	48.16	256	-128 to +127
12	72.25	4096	-2048 to +2047
16	96.33	65,536	-32,768 to +32,767
20	120.41	1,048,576	-524,288 to +524,287

Note: quantisation is not the only source of noise/error

Quantisation

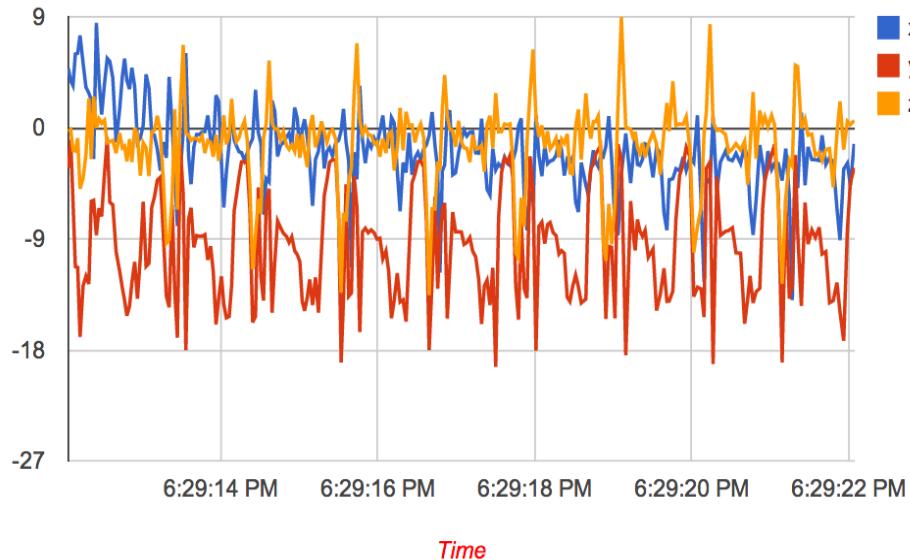
common audio quantisation bit depths:

bit depth	application
8	telephone
8 or 12	voice over IP
16	audio CD
16, 20, or 24	Blu-ray audio track

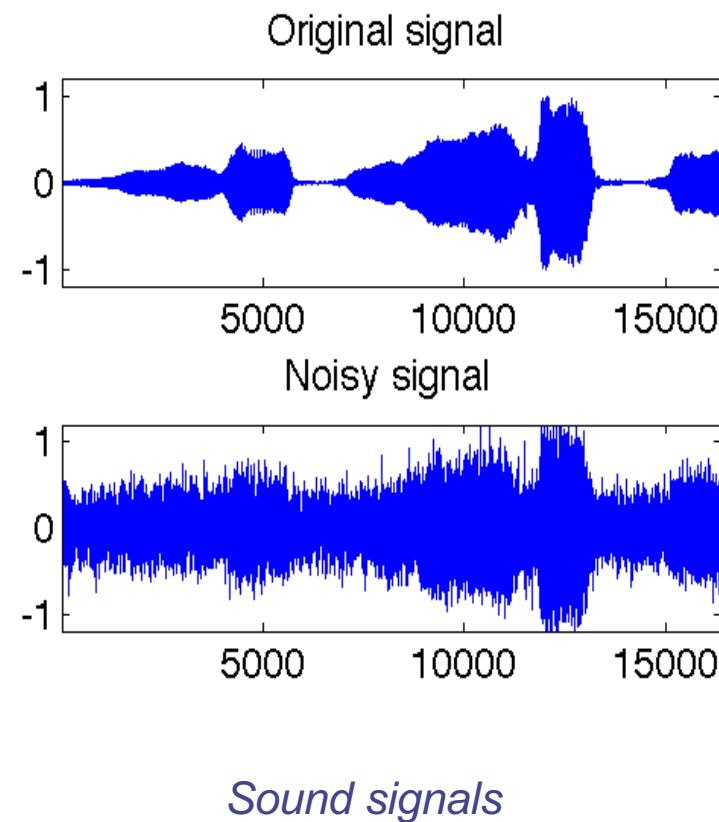
common video quantisation bit depths (colour depths):

bit depth	application
6×3	high colour
8×3	true colour
$10, 12, \text{ or } 16 \times 3$	deep colour

Noisy Sensor Signals



Typical pattern of x-, y-, and z accelerations while walking with smartphone in pocket.



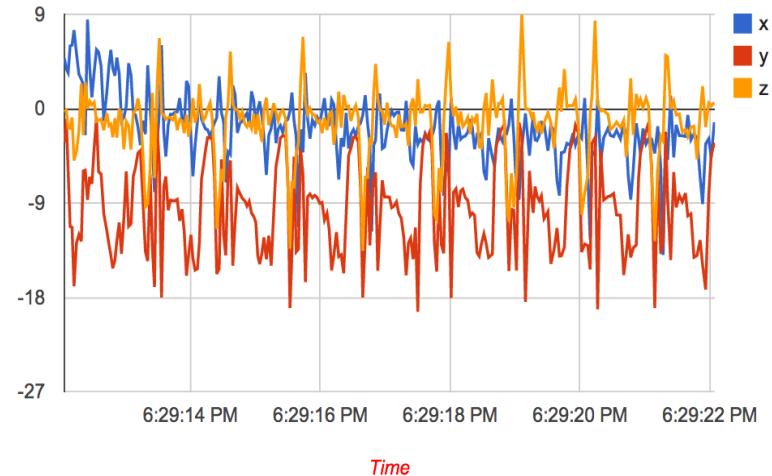
Sound signals

Time-series smoothing

- Moving average smoothing

- $s_t = (x_t + x_{t+1} + x_{t+2})/3,$

- Window size: 3

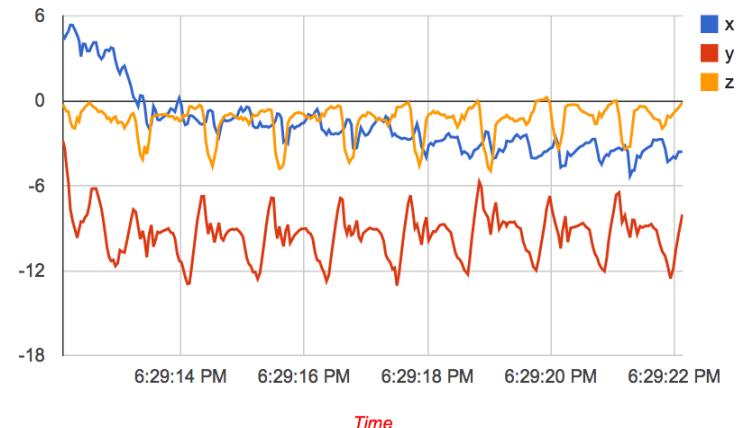


- Exponential smoothing

$$s_1 = x_0$$

$$s_t = \alpha x_{t-1} + (1 - \alpha)s_{t-1} = s_{t-1} + \alpha(x_{t-1} - s_{t-1}), t > 1$$

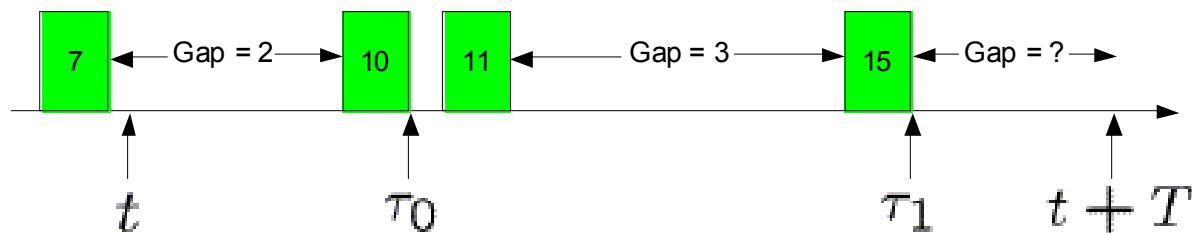
- where α is the smoothing factor, and $0 < \alpha < 1$



$$\alpha = 1/6$$

Link quality estimation

- How to estimate, on-line, in the field, the actual link quality?
- Requirements
 - Precision – estimator should give the statistically correct result
 - Agility – estimator should react quickly to changes
 - Stability – estimator should not be influenced by short aberrations
 - Efficiency – Active or passive estimator



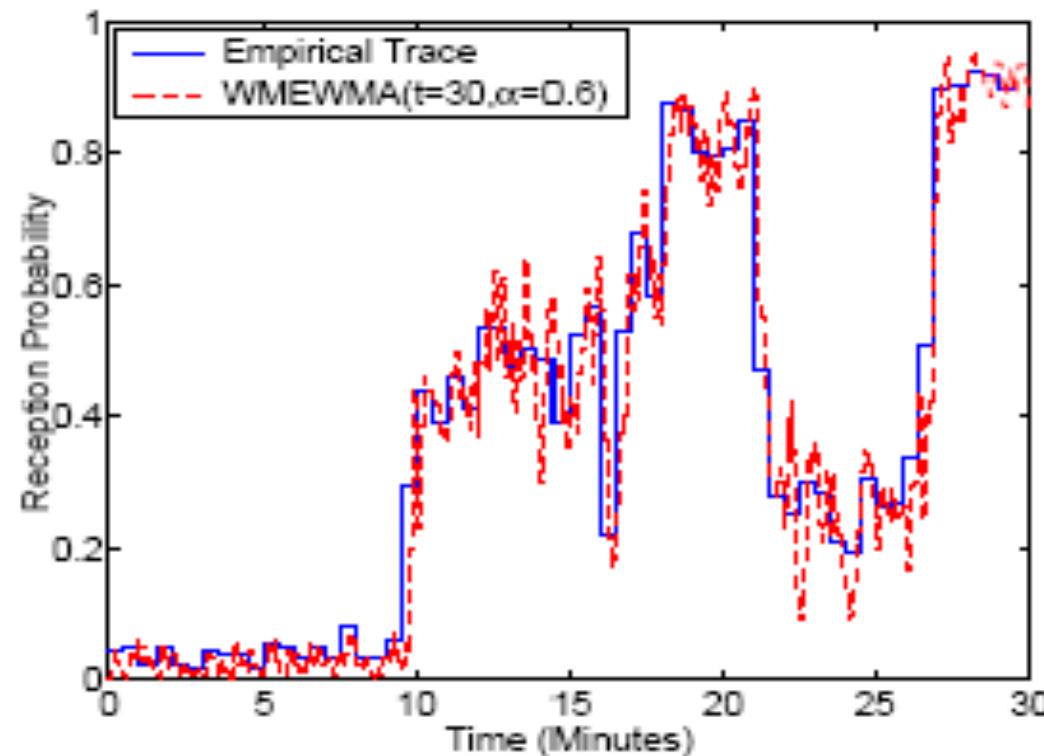
- Example:
WMEWMA
only estimates
at fixed intervals

$$P_n = \alpha P_{n-1} + (1 - \alpha) \frac{r_n}{r_n + f_n}$$

r_n : received packets in interval
 f_n : packets identified as lost

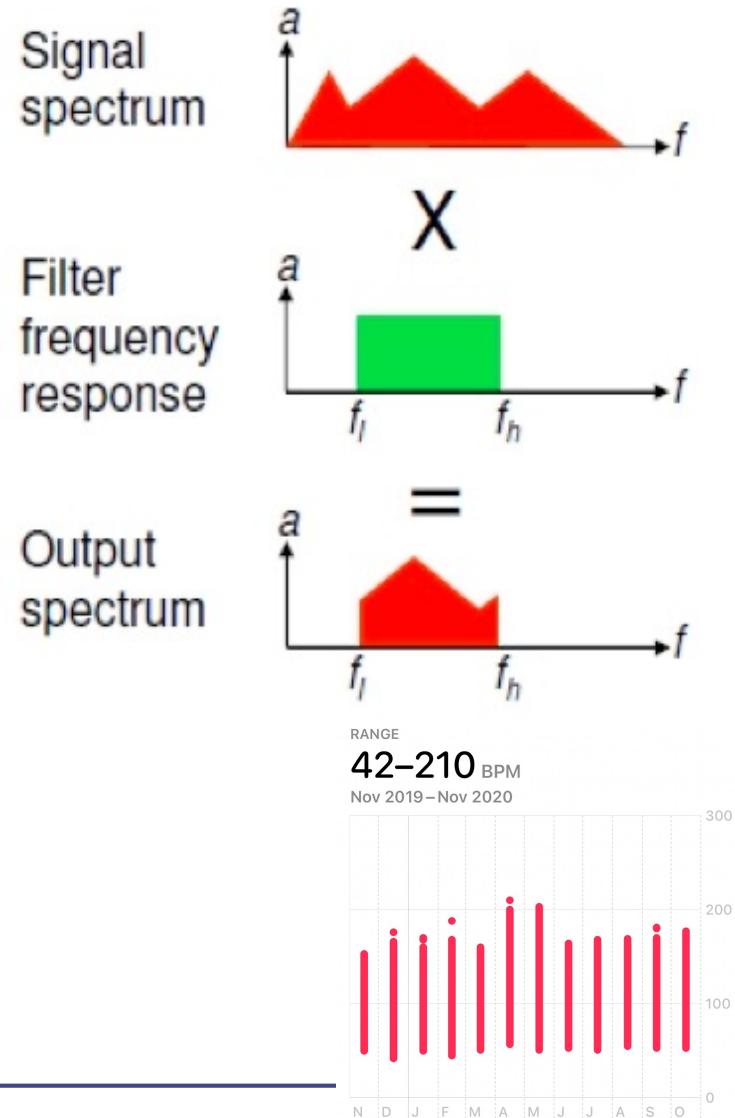
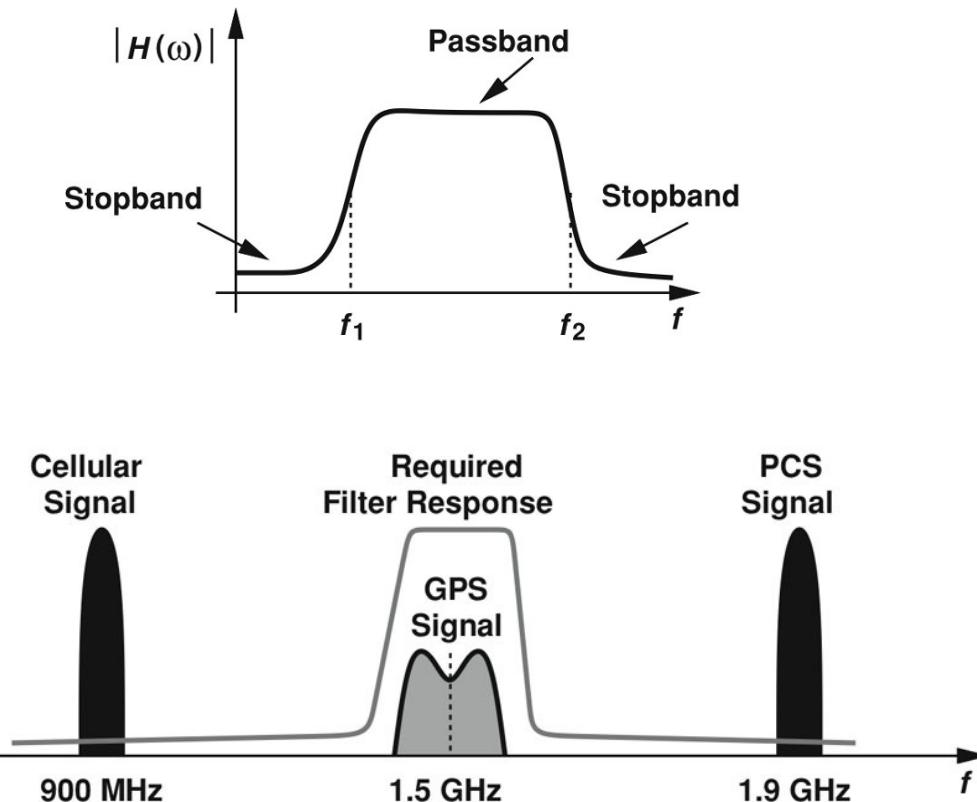
Link estimation (2)

- Method used in the paper
 - Snooping
 - Assumption: Each node sends periodic traffic
- Used WEMA estimator



Frequency-domain filtering

- Bandpass filtering



Finding the frequency of a signal...

- Discrete Fourier Transform – $O(n^2)$
- Fast Fourier Transform – $O(n\log(n))$
- Mel-frequency Cepstral Coefficients (MFCC)

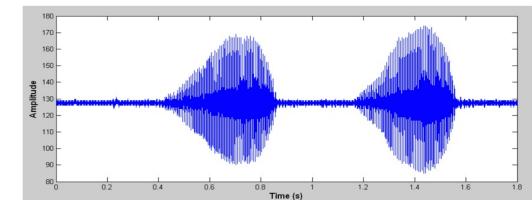
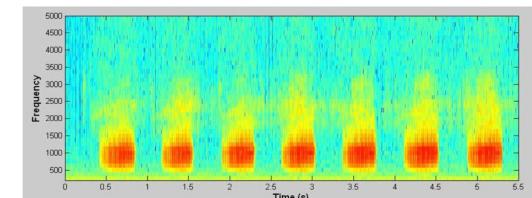


Fig. 2. The waveform graph of *Cyclorana cryptotis*.



a *cryptotis*.

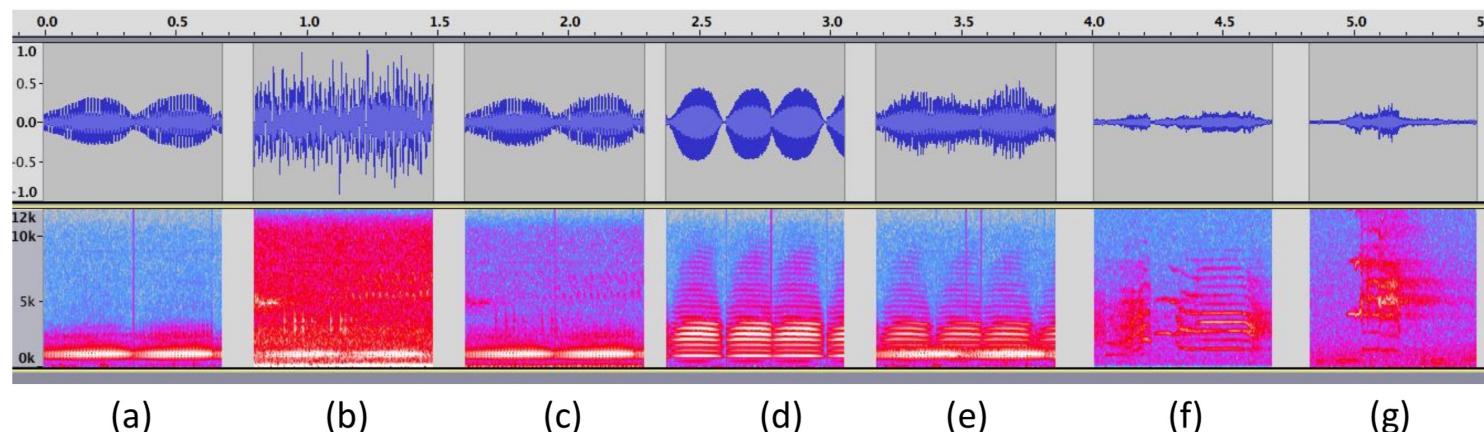


Figure 5: (Picture view best in color) (a) *C. Cryptotis*' call without noise, (b) *C. Cryptotis*' call with noise (0dB SNR), (c) *C. Cryptotis*' call with noise (10dB SNR), (d) *C. Cultripes*' call without noise, (e) Combined sound from (a) and (d) (0dB SSR), (f) Typical cockatoo sound from experiment, (g) Typical rainbow lorikeet call from experiment

Signal Processing in Embedded Systems

challenges:

Any embedded application of integrated circuits seeks to minimise, simultaneously, four factors:

- the **number of transistors** employed, which impacts die and package size, unit cost and power consumption;
- the **number of clock cycles** required, which impacts performance and power consumption;
- the **time taken to develop the application**, which strongly influences its market acceptance;
- **nonrecurring engineering (NRE) costs** such as mask manufacturing and the cost of hardware and software development.

Signal Processing in Embedded Systems

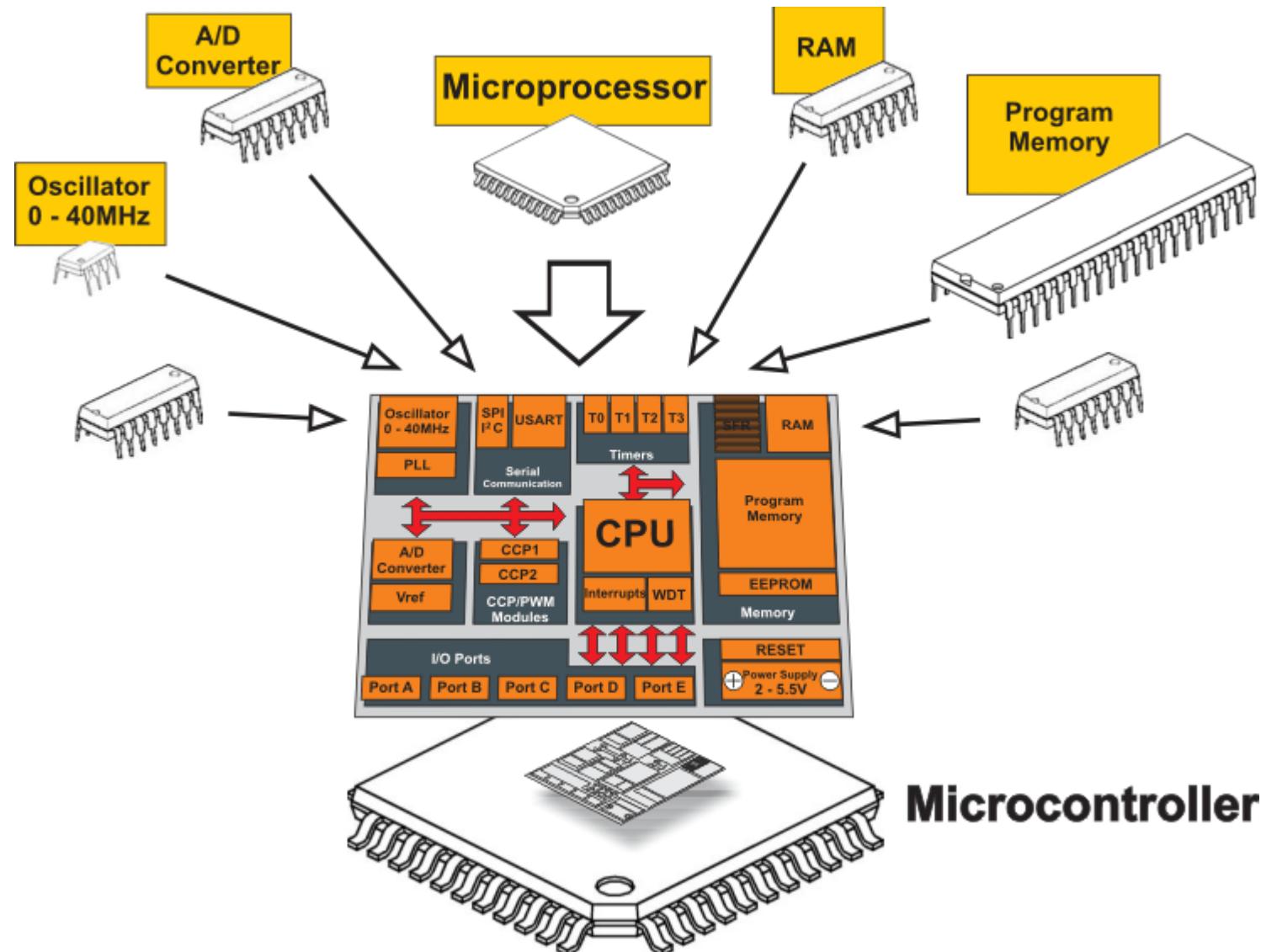
existing solutions to address the challenges:

- microcontroller (MCU)
- digital signal processor (DSP)
- field programmable gate array (FPGA)
- application-specific integrated circuit (ASIC)
- application-specific standard product (ASSP)
- configurable processor (CP)
- reduced instruction set computer / general-purpose processor (RISC/GPP)
- graphics processing unit (GPU)
- tensor processing unit (Google)/Myriad VPU (Intel)

Microcontroller (MCU)

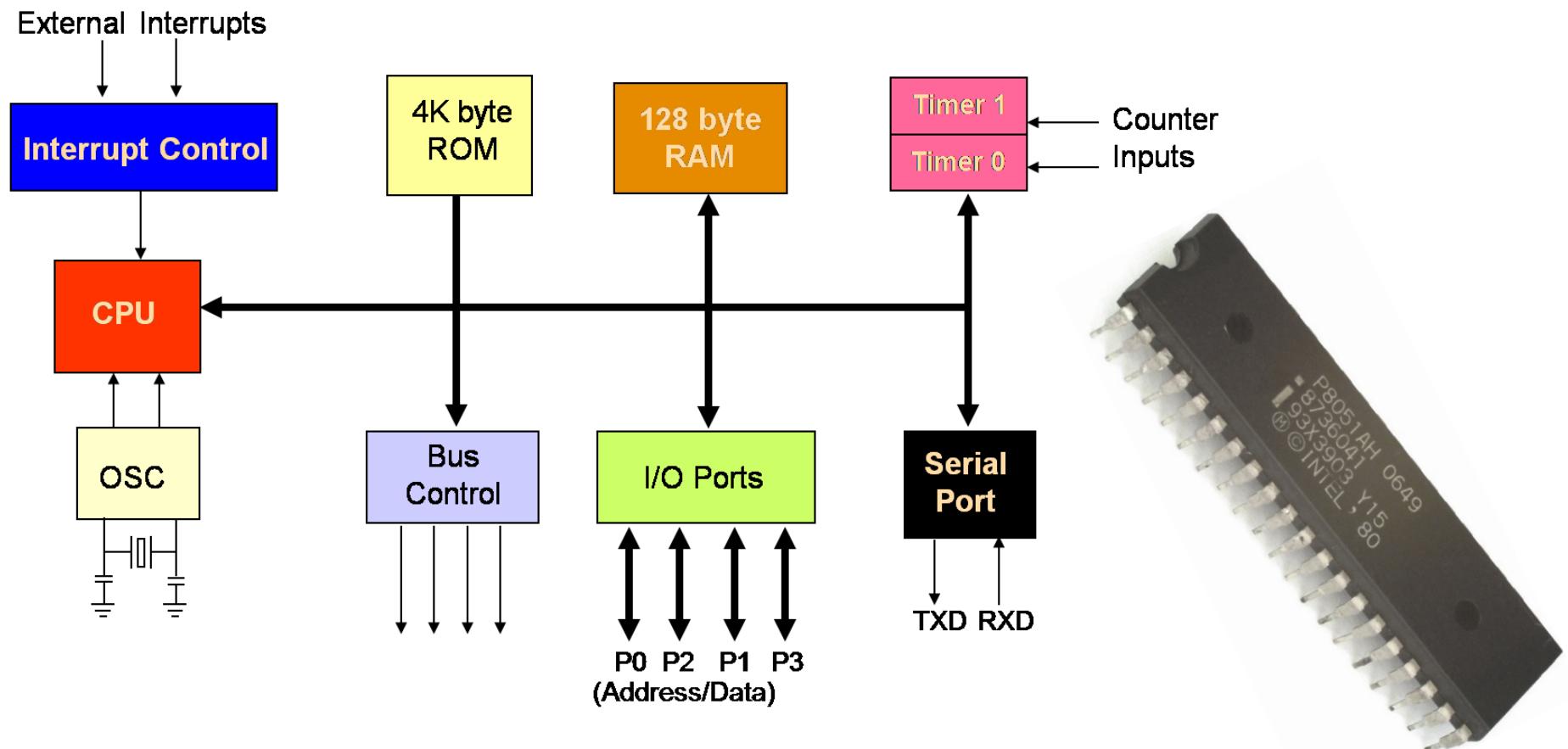
- MCUs are **general-purpose** devices for information processing and control that can be adapted to a wide variety of applications by software.
- Application development effort is limited to **software** development and validation, and NRE costs are amortised amongst all the users of a particular MCU architecture.
- Clock cycle optimisation is determined by **code optimisation**, and the code footprint influences the number of transistors required for memories.
- **Compact code** that makes the most efficient use of the MCU architecture is essential.
- MCUs generally use transistors and clock cycles efficiently, but not optimally.

Microcontroller (MCU)



Microcontroller (MCU)

The 8051 Block Diagram



Microcontroller (MCU)

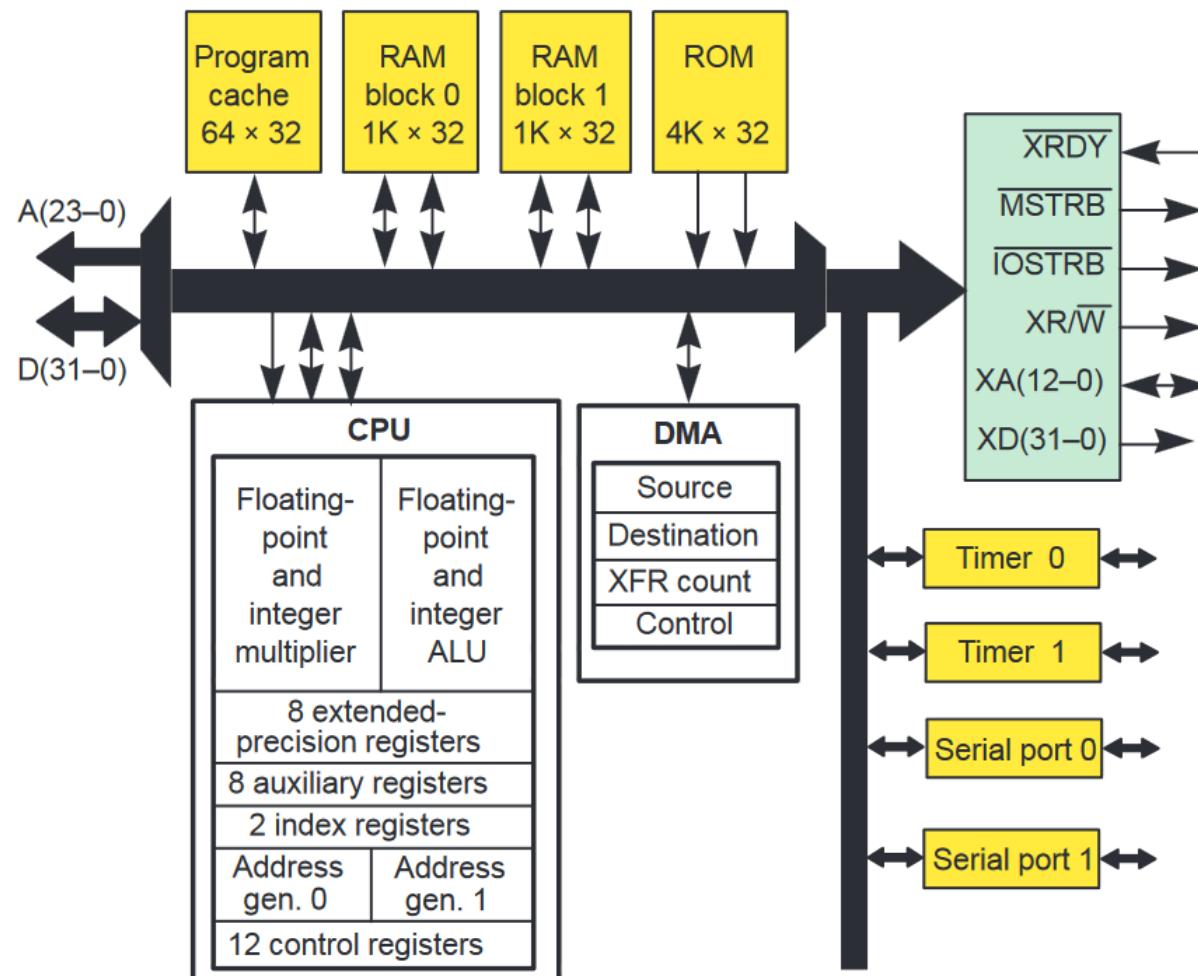


Digital Signal Processor (DSP)

- DSP is a **specialised** microprocessor with its architecture optimised for the operational needs of digital signal processing tasks.
- DSPs **hard-wire** the basic functions of many signal-processing algorithms.
- This **optimises transistor use and clock cycles** for the required operations, at the expense of flexibility.
- DSPs are optimised for streaming data and use special memory architectures that are able to **fetch multiple data or instructions**.
- **Code is simpler** than that required for MCUs. In many cases a DSP is an optimal solution for some but not all of the functions required by an application.
- Many MCUs include basic DSP operations in their instruction set so they can do simple signal processing without a dedicated DSP.

Digital Signal Processor (DSP)

TMS320C30 Block Diagram



Field Programmer Gate Array (FPGA)

- A FPGA is an integrated circuit designed to be **configured** by a customer or a designer after manufacturing.
- The FPGA configuration is generally specified using a hardware description language (HDL).
- FPGAs have large resources of logic gates and RAM blocks to implement complex digital computations.
- FPGAs limit development effort to the **coding** required to configure them;
- share NRE costs amongst a very large population of users,
- at the expense of a high level of **transistor redundancy** and therefore **high unit costs**
- and a **limited optimisation of clock cycles**.
- **Power consumption** is far from optimal.

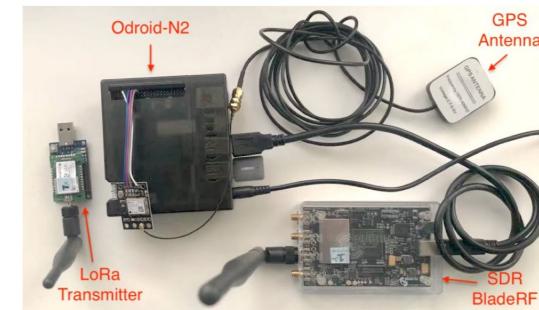


Figure 8: *Nephelai* gateway and a LoRa transmitter

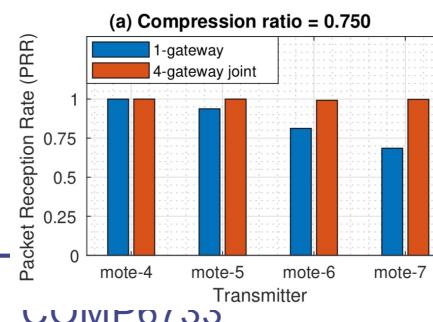
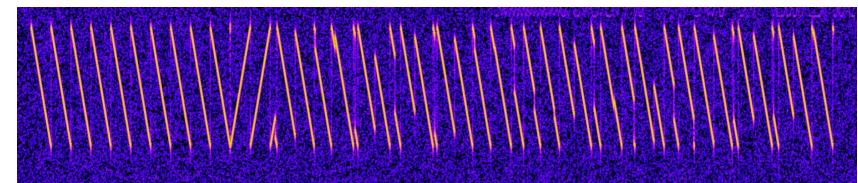
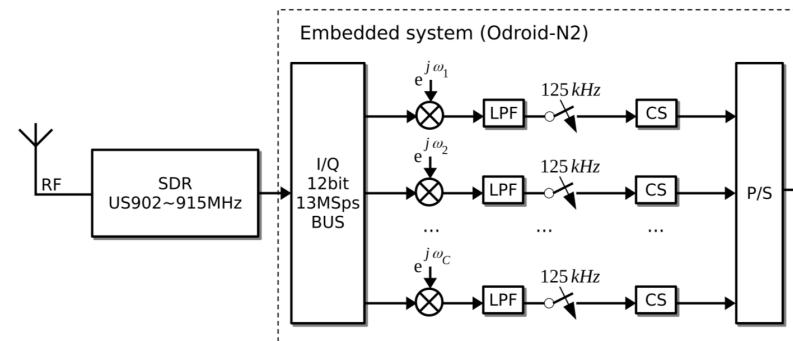
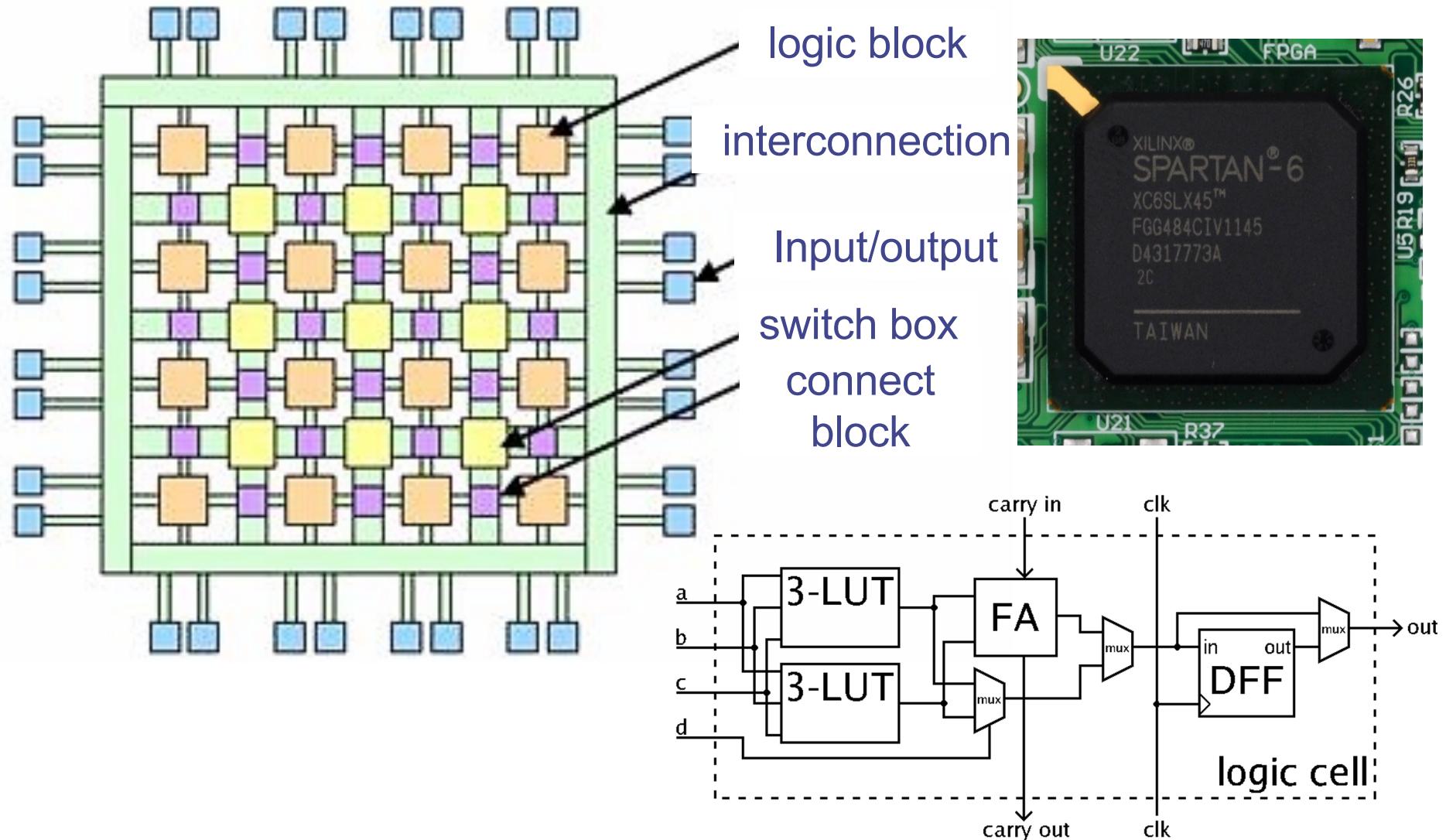
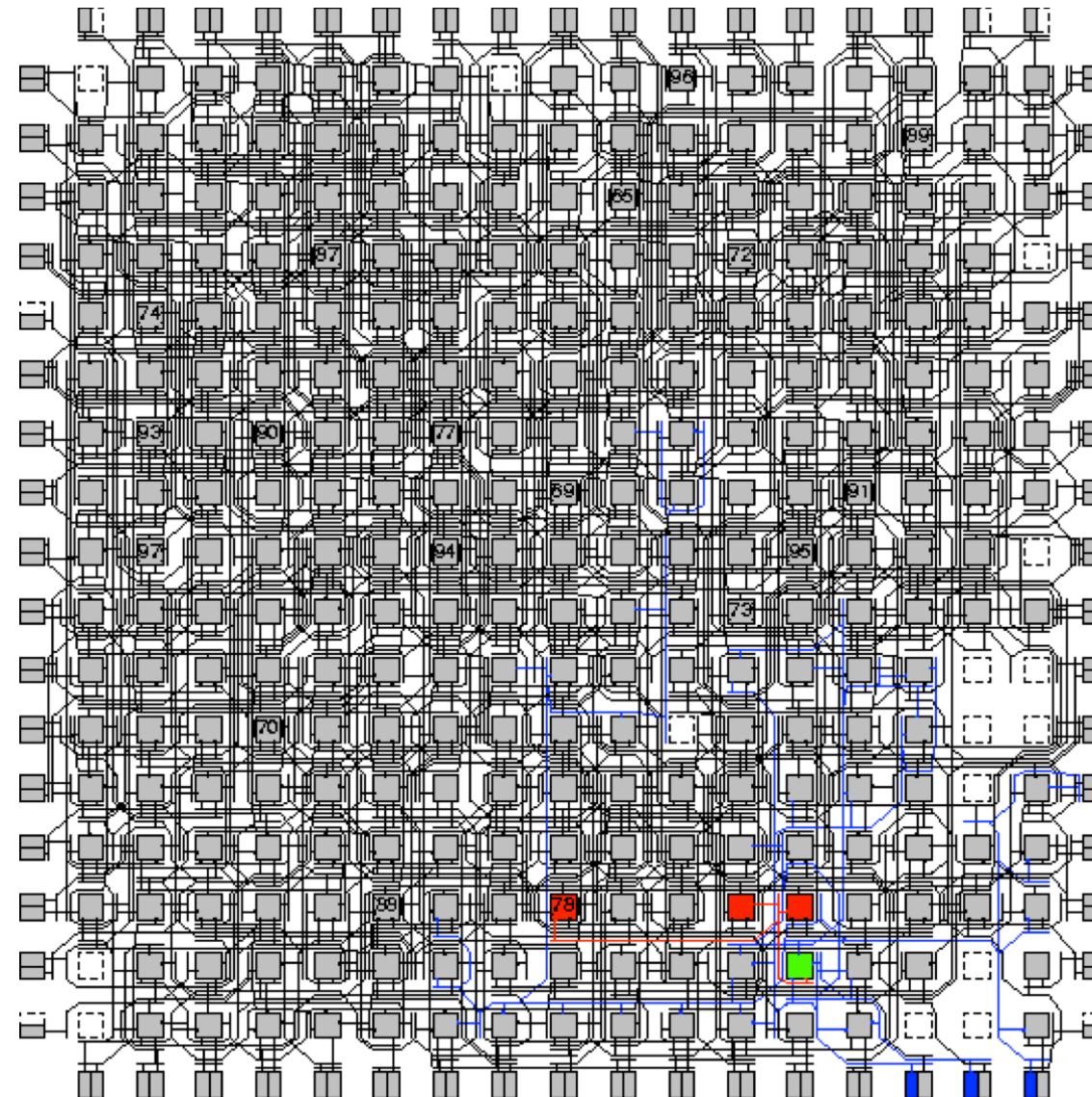


Figure 12: Outdoor evaluation. Red squares marked with A/B/C/D and antenna direction indicate the APs. Green circles indicate the ground-truth of transmitter's locations.

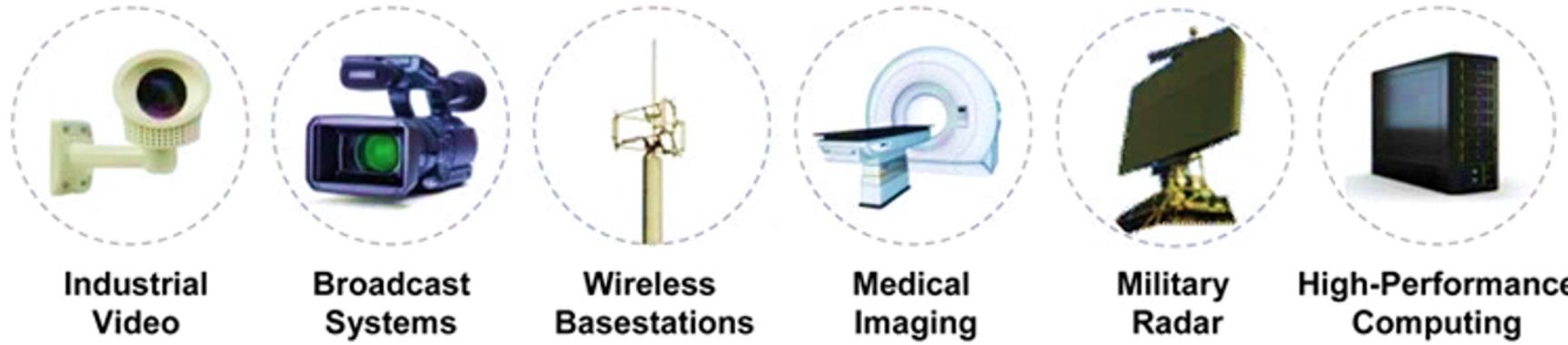
Field Programmer Gate Array (FPGA)



Field Programmer Gate Array (FPGA)

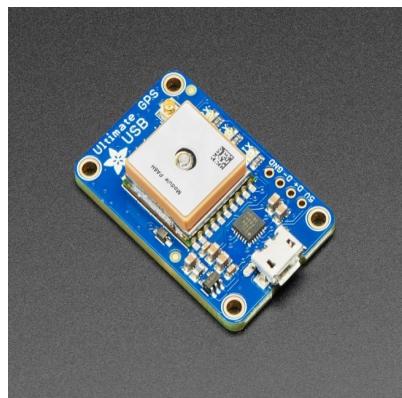


Applications of DSP, FPGA



Application-Specific Integrated Circuit (ASIC)

- MCUs, DSPs, and FPGAs are delivered as **standard products**, generally with a wide range of options, but nevertheless for any application the closest match always contains some redundant transistors and input/outputs.
- Application-specific functions, in particular analogue operations, must often be implemented off-chip.
- Therefore, in MCUs, DSPs, and FPGAs, die size, package size, pinout and power consumption are **less than optimal**.



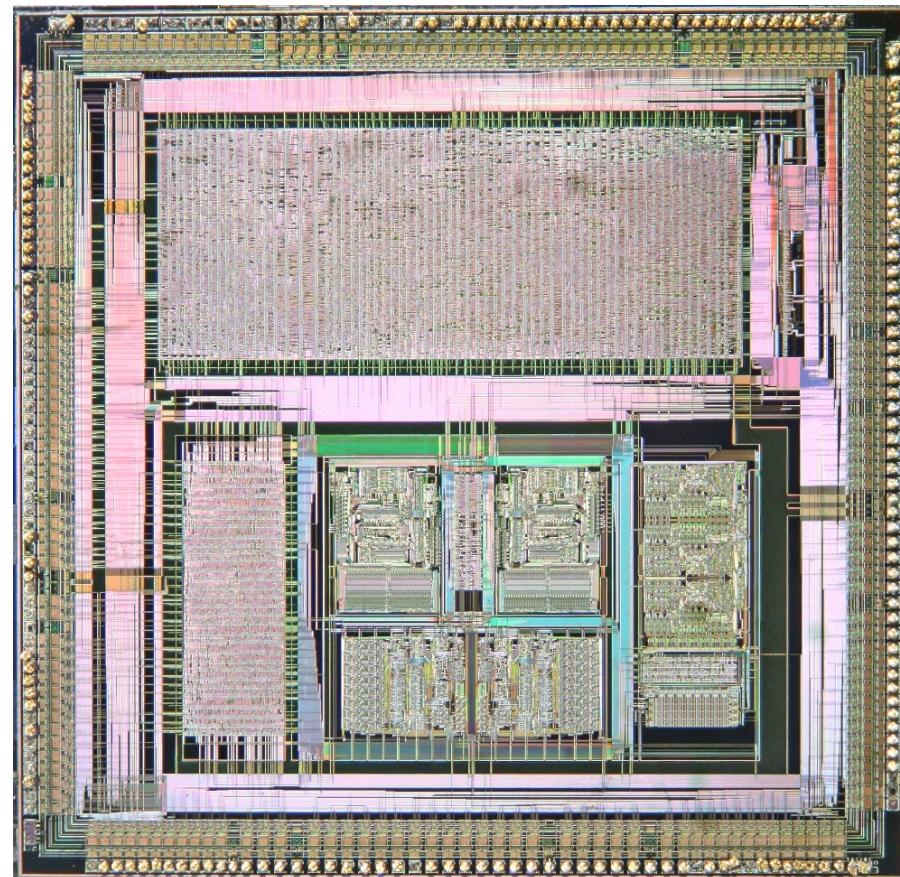
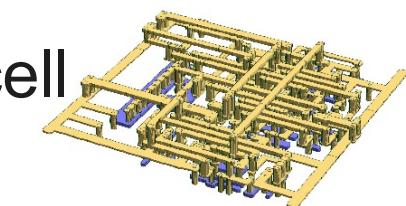
Application-Specific Integrated Circuit (ASIC)

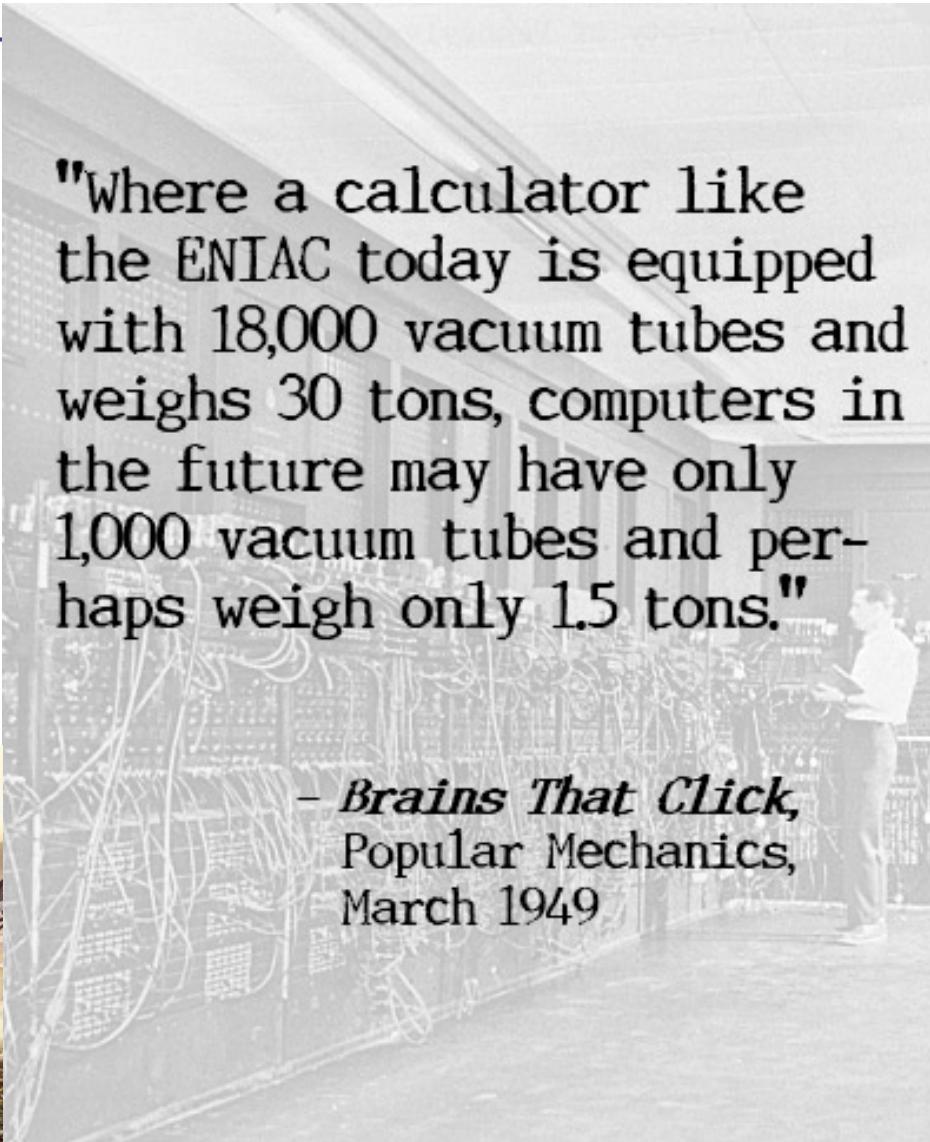
- An ASIC is custom-designed for a **particular application**, possibly embedding one or more MCU or DSP cores, with as much as possible of the total system functionality implemented on a single die.
- This **optimises the number of transistors and clock cycles** and therefore unit cost and power consumption,
- at the expense of **development time** and **NRE cost** that are generally an order of magnitude higher than those for MCUs, DSPs or FPGAs.
- The NRE cost of an ASIC can run into the millions of dollars.

Application-Specific Integrated Circuit (ASIC)

design options:

- standard-cell
 - gate-array
 - full-custom
 - structured





— *Brains That Click*,
Popular Mechanics,
March 1949



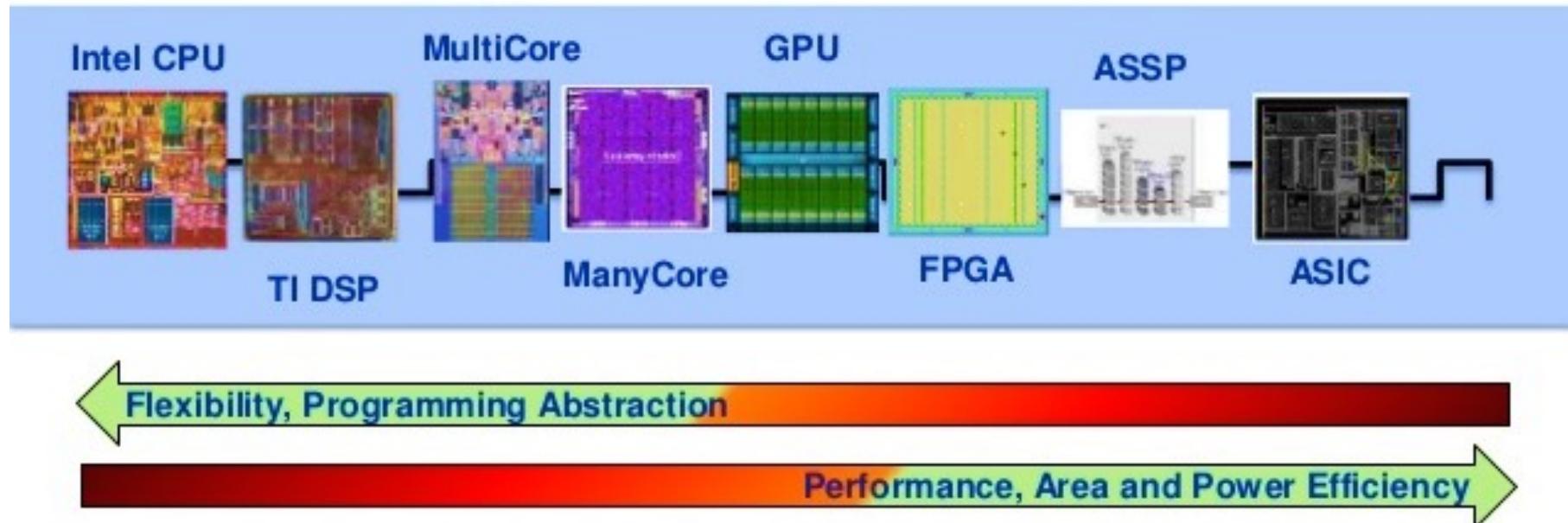
Trade-offs

- The four solutions represent different trade-offs towards achieving the four optimisations.
- The choice for any particular application is an engineering **compromise**.
- In most cases, the choice depends on a complex combination of factors, and no single technology is ideal.
- Different technology mixes are often most appropriate at different stages of the **lifecycle** of the end-user product.
- During prototyping and production ramp-up an FPGA or MCU/DSP-plus-FPGA solution may be preferable, in order to reduce development time and cost.

Trade-offs

- When the product goes into **high volume**, its functionality can be re-mapped into an ASIC that embeds the MCU or DSP core from the standard product, and absorbs the logic from the FPGA,
- thereby **optimising** die size, unit cost, clock cycles and power consumption without the need to rewrite the software.
- The high NRE costs associated with ASIC development are amortised over the high production volume.

Trade-offs



CPU:

- Market-agnostic
- Accessible to many programmers (C++)
- Flexible, portable

FPGA:

- Somewhat Restricted Market
- Harder to Program (Verilog)
- More efficient than SW
- More expensive than ASIC

ASIC

- Market-specific
- Fewer programmers
- Rigid, less programmable
- Hard to build (physical)

<http://www.altera.com>

Trade-offs

	time to market	performance	price	development ease	power consumption	feature flexibility
MCU	excellent	fair	excellent	good	fair	excellent
DSP	excellent	excellent	good	excellent	excellent	excellent
FPGA	good	excellent	poor	excellent	poor	good
ASIC	poor	excellent	excellent	fair	good	poor
ASSP	fair	excellent	good	fair	excellent	poor
CP	poor	excellent	good	poor	good	fair
RISC	good	good	fair	good	fair	excellent

<http://www.ti.com>

References

- Optional readings:
 - James D. Broesch, (2009), , Digital Signal Processing, Newnes, Burlington.