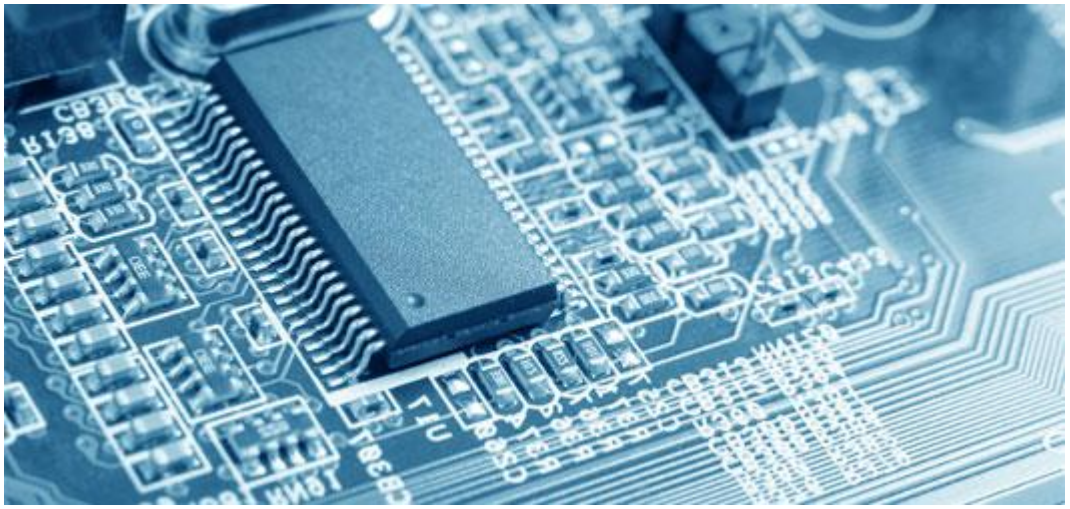
NAZARBAYEV UNIVERSITY
SCHOOL OF SCIENCE AND TECHNOLOGY



CSCI502 – Hardware/Software Co-Design

Self-Study Lecture 4 – Hardware for Embedded Systems: Cashe, Peripheral Interfacing, Analog-Digital Conversion

21 January 2019

Course Logistics

Reference Reading:
Real-Time Embedded Systems: Chapter 2
Logic and Computer Design Fundamentals. 5th edition: Chapters 5,11,12 (slide relevant material)
Exploring BeagleBone. 2nd edition: Chapter 4

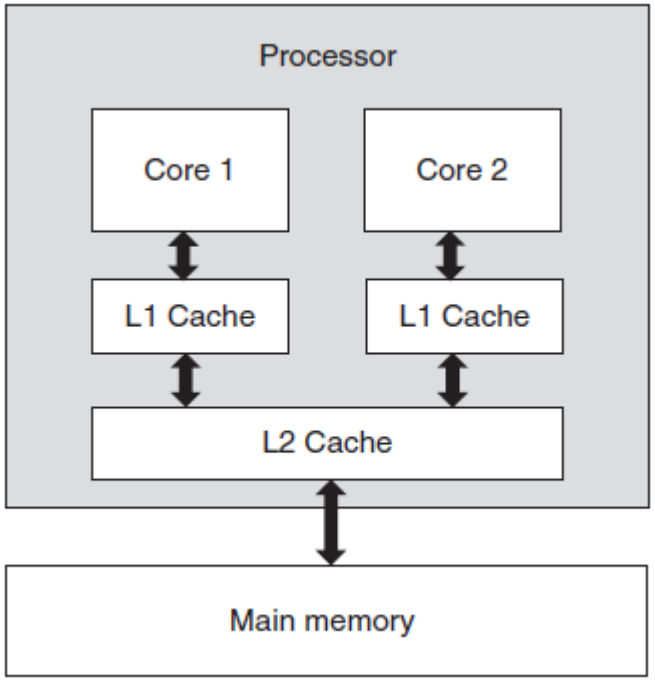
Introduction

- Embedded systems interact closely with input and output devices including sensors and actuators
- Hardware structure can affect timing and require special software design techniques

Cashe

- A cache – lowest level of the hierarchy – a small block of fast memory where frequently used instructions and data are kept. The cache is much smaller than the main memory
- Main memory: RISC serves directly most of CPU instructions, operand fetches not satisfied by the cache

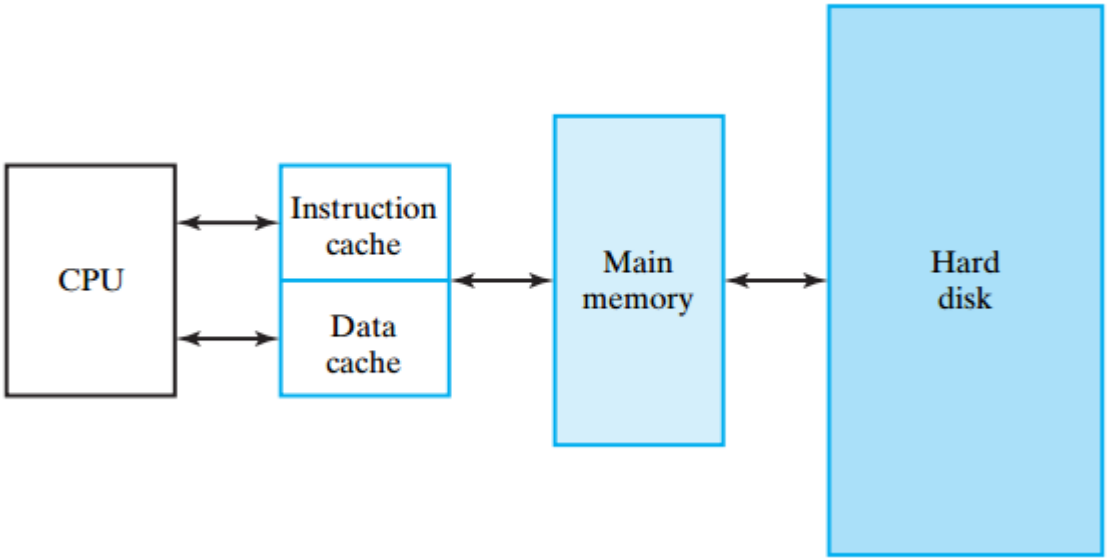
- Some CPUs have multilevel cache memory. They are referred to as L1 for level one, L2 for level two, and so on.
- A CPU directly works with L1 cache, so it has data to work on every processor cycle at the best.
- The L2 cache is expected to feed data to the L1 cache every few processor cycles.



A dual-core processor with two levels of cache memory

Cache

- A small block of fast memory where frequently used instructions and data are kept. The cache is much smaller than the main memory.
- Suppose that cache memory is divided to two cashes: one for instructions and one for data.
- Thus, one instruction can be fetched and one operand can be fetched/stored, in a single clock cycle if the cashes are fast enough



Cache

- Assume that typical access time to DRAM is about 10 ns
- If CPU accesses only RAM memory than CPU with 1 ns clock cycle would operate too slow. At 1/10 of its full speed
- If 95% of all memory accesses are made with cashes taking 2 ns and
- Remaining 5% of memory accesses take 10 ns.
- Then the average time is **0.95 x 2 + 0.05 x10 = 2.4 ns**

Cache

- Performance benefits are a function of cache hit ratio.
- If needed data or instructions are not found in the cache, then the cache contents need to be written back (if any were altered) and overwritten by a memory block containing the needed information.
- Overhead can become significant when the hit ratio is low.
  - Therefore a low hit ratio can degrade performance.
- If the locality of reference is low, a low number of cache hits would be expected, degrading performance.
- Using a cache is also non-deterministic – it is impossible to know *a priori* what the cache contents and hence the overall access time will be.
- In multitasking real-time systems – high probability of cache misses due to frequent switching between different tasks and interrupts.

Memory Technologies: ROM

Two classes of memory :
RAM – random access memory
ROM – read-only memory

- Electrically erasable programmable ROM (EEPROM) and Flash (ROM) can be rewritten similar to RAM devices
- Erasing and writing process is much slower comparing to RAM and limited to 100000 -1000000 times
- EEPROM is mainly used for nonvolatile program and parameter storing
- Flash – application program and data records

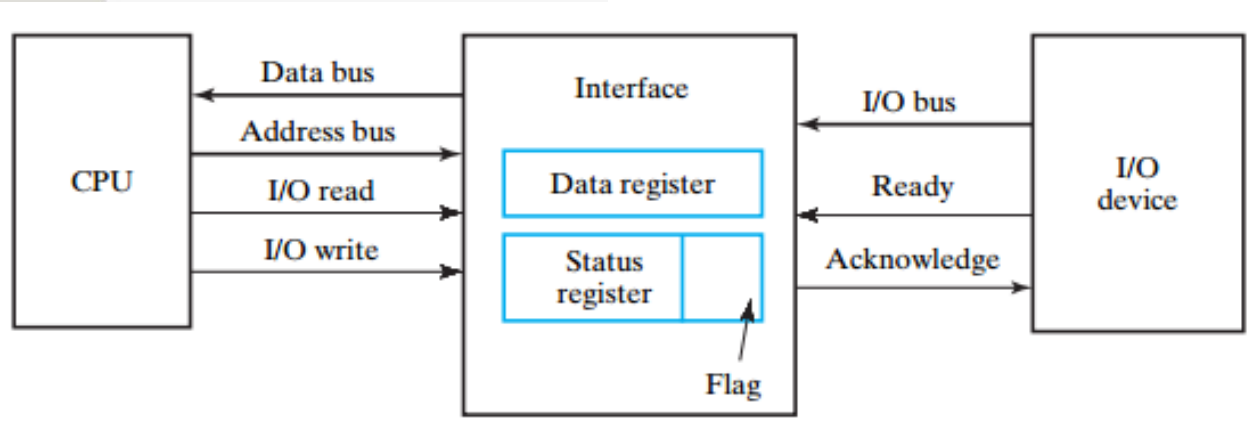
Peripheral Interfacing

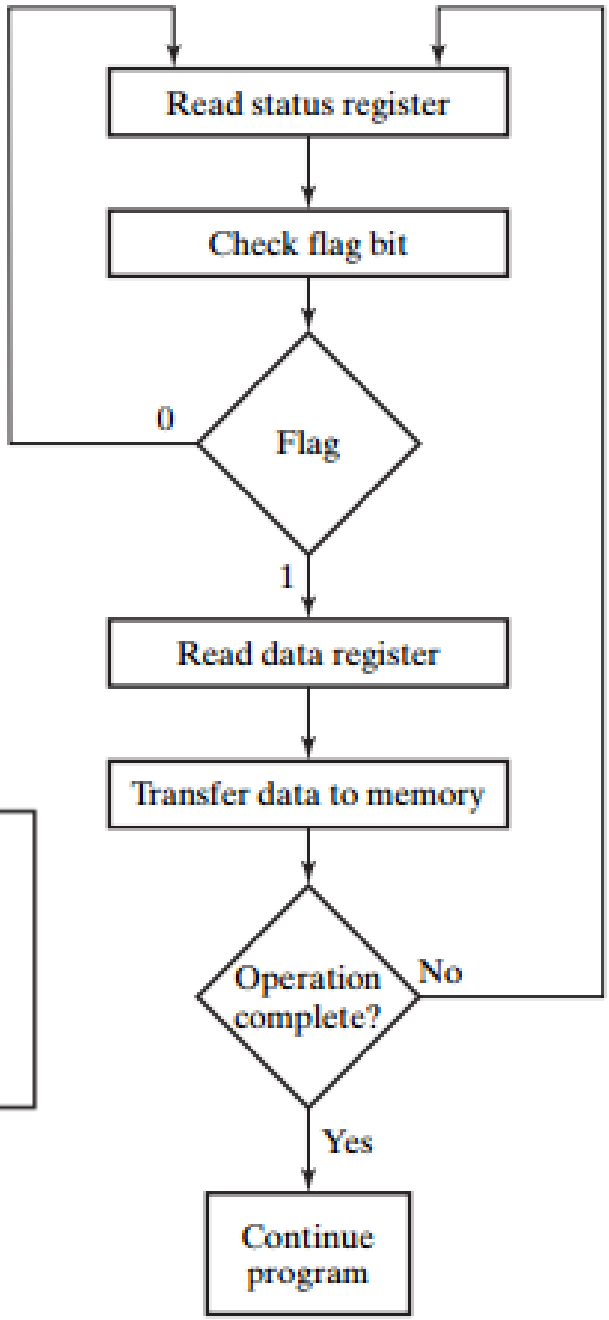
- Peripheral, sensor and actuator interfacing developing much slower than memory and processor architectures
- Fundamental practices for I/O handling are the same:
  - Polled I/O (program-controlled transfer)
  - Interrupt-driven I/O
  - Direct memory access

- **In polled I/O system** the status of I/O device is checked periodically, or regularly.
- Advantage - simplicity
- Disadvantage – CPU loading with unnecessary status requests

Polled Loop Systems

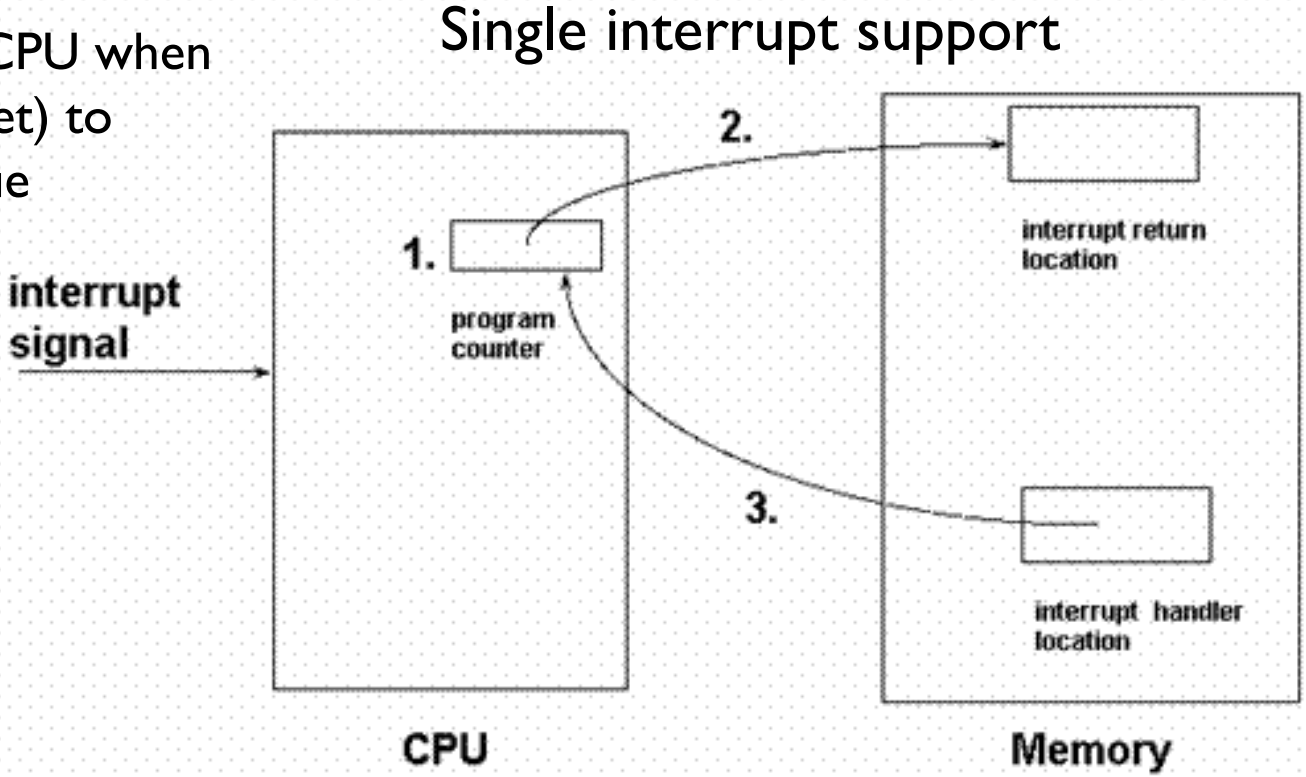
- CPU check the flag to define whether there is new byte in the interface data register or not.
- Assume the input device transmits data at 100 bytes/s or 1 byte every 10 msec.
- If CPU checks the flag in 100 nsec than CPU will check the flag 100000 times between each transfer





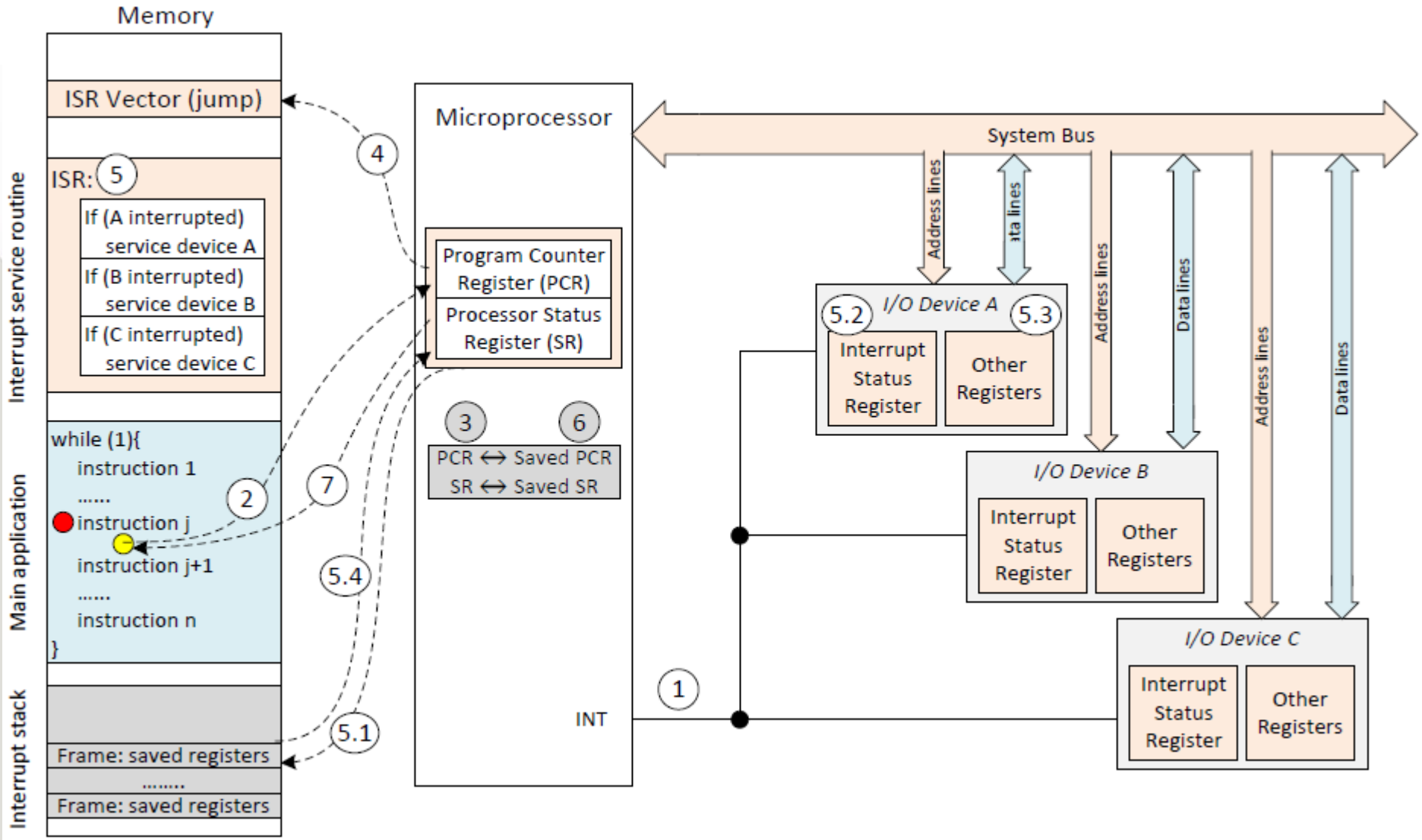
Interrupt Driven Input/Output

- Interface informs CPU when it is ready (flag is set) to transfer data – issue interrupt request



Step 1: finish the currently executing macroinstruction.
Step 2: save the contents of the program counter to the interrupt return location.
Step 3: load the address held in the interrupt handler location into the program counter. Resume the fetch and execute sequence.

Non-vectored Interrupting



Non-vectored Interrupting

1. CPU executes an instruction *j* of a user program, and one of the I/O devices issues an interrupt request (IRQ) signal to CPU's interrupt input pin (INT).
2. CPU detects the IRQ and completes the instruction *j*. The value of the program counter register (PCR) is the location of the next instruction of the user program.
3. The status register (SR) and the PCR are saved.
4. The PCR is loaded with the memory location of the common interrupt vector, which is typically a jump instruction, pointing to the start location of the interrupt service routine (ISR).

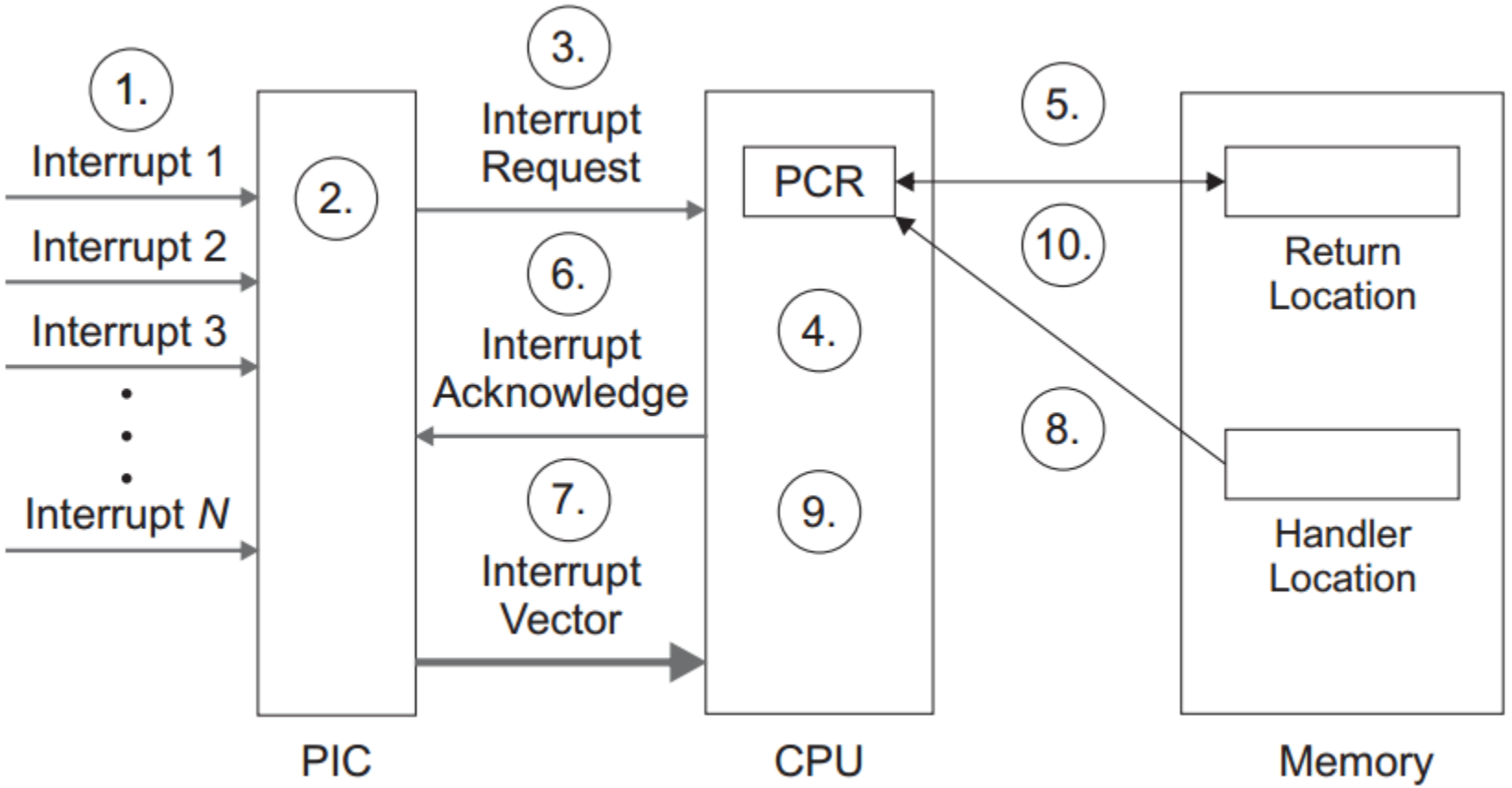
5.1. Inside the ISR, the commonly used registers are first saved onto the interrupt stack.
5.2. CPU searches through the I/O devices (checking interrupt status registers), from high to low priority, to identify the interrupt source (requesting device).
5.3. The portion of code pertinent to the requesting device is executed, which typically entails the access of the ports (registers) on the device.
5.4. At the end of the ISR, the top frame of the interrupt stack is popped up and the context of the user task is restored.

6. The original PCR value is restored.
7. The processor is ready to run the next instruction of the user program.

Vectoring Interrupting: Programmable (Priority) Interrupt Controller (PIC)

1. The PIC receives several simultaneous interrupt requests
2. The PIC processes first the request with highest priority
3. The CPU receives an interrupt request from the PIC
4. The CPU completes the currently executing instruction
5. The CPU stores the content of the PC to memory
6. The CPU acknowledges the interrupt to the PIC
7. The PIC sends the interrupt vector to the CPU
8. The CPU loads the corresponding interrupt- handler address to the PC
9. The CPU executes the interrupt routine
10. The CPU reloads the original program counter content (PCR) from memory

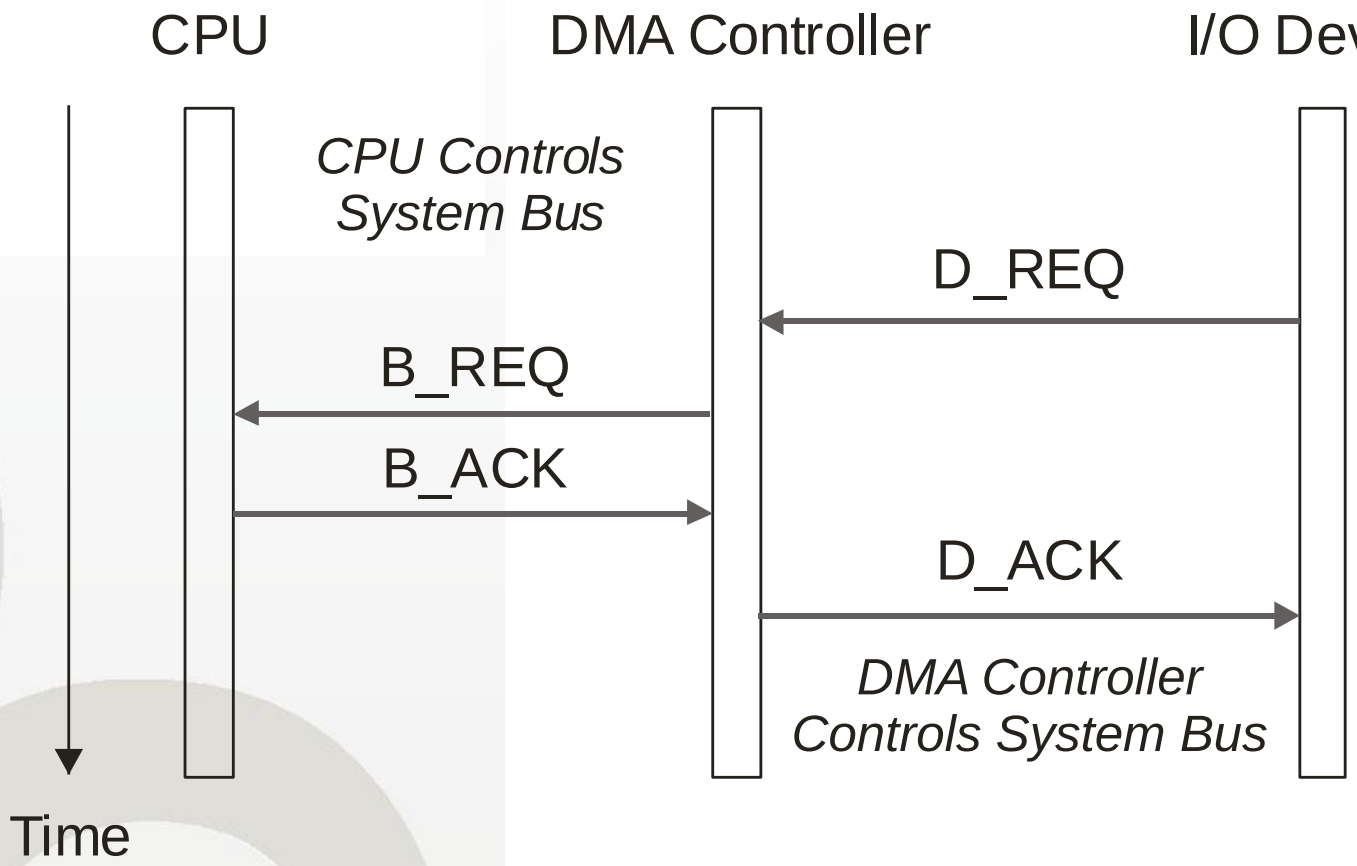
Priority Interrupt Controller (PIC)



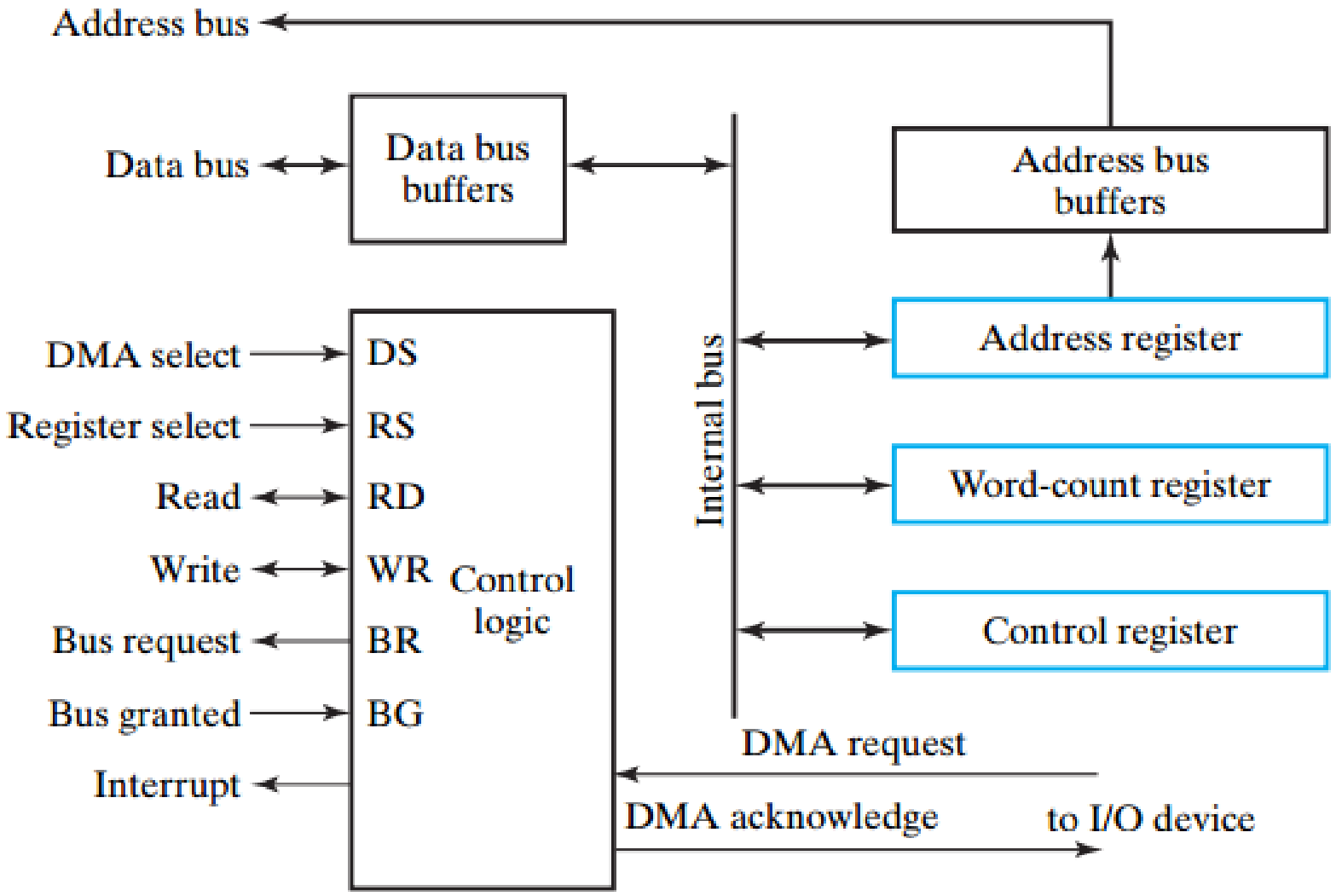
Direct Memory Access (DMA)

- Access to the computer's memory is given to other devices in the system without CPU intervention.
- Information is deposited directly into main memory by the external device.
- DMA controller is required unless the DMA circuitry is integrated into the CPU.
- Because CPU participation is not required, data transfer is fast.

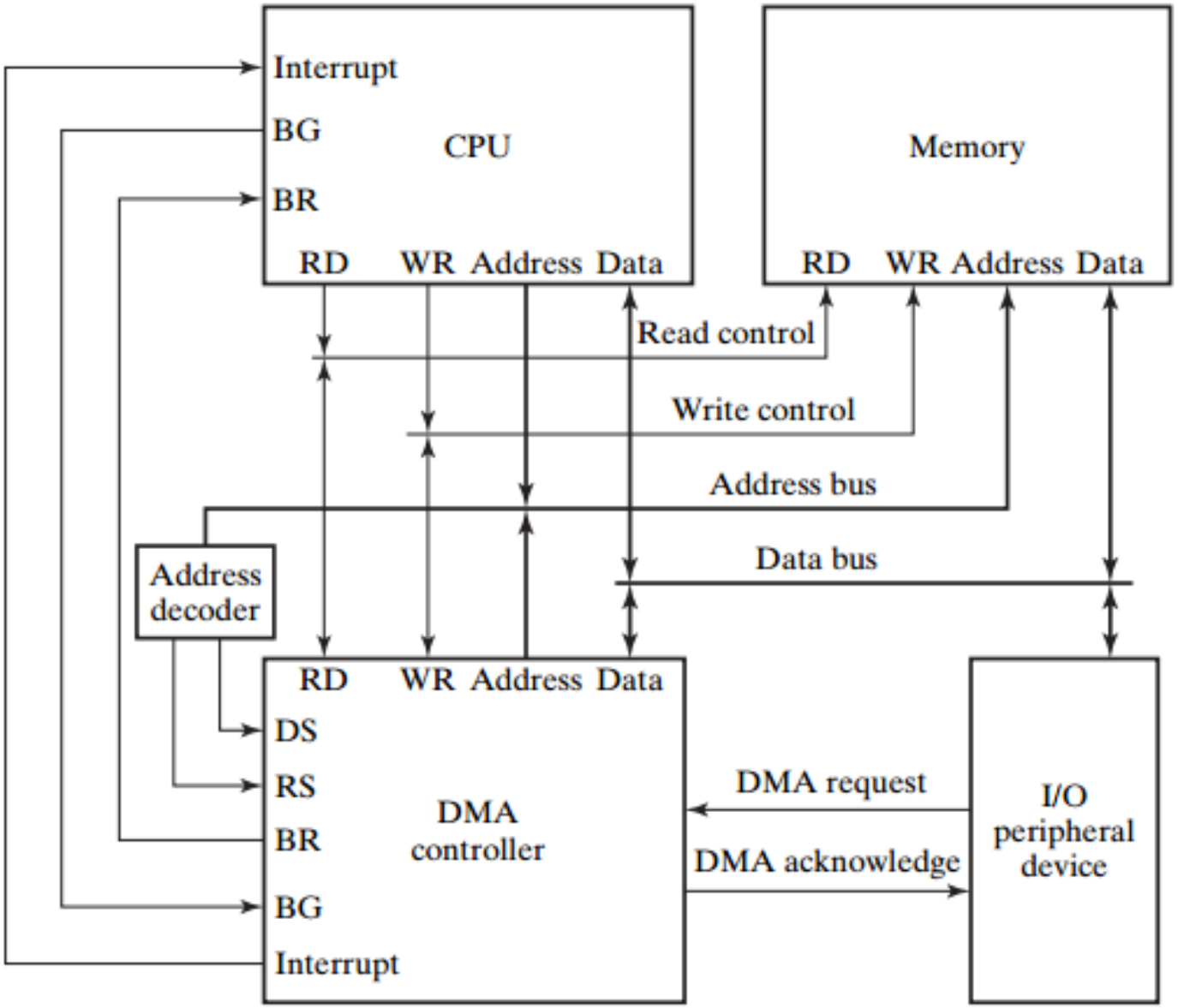
DMA Handshaking



DMA Controller



DMA Transfer

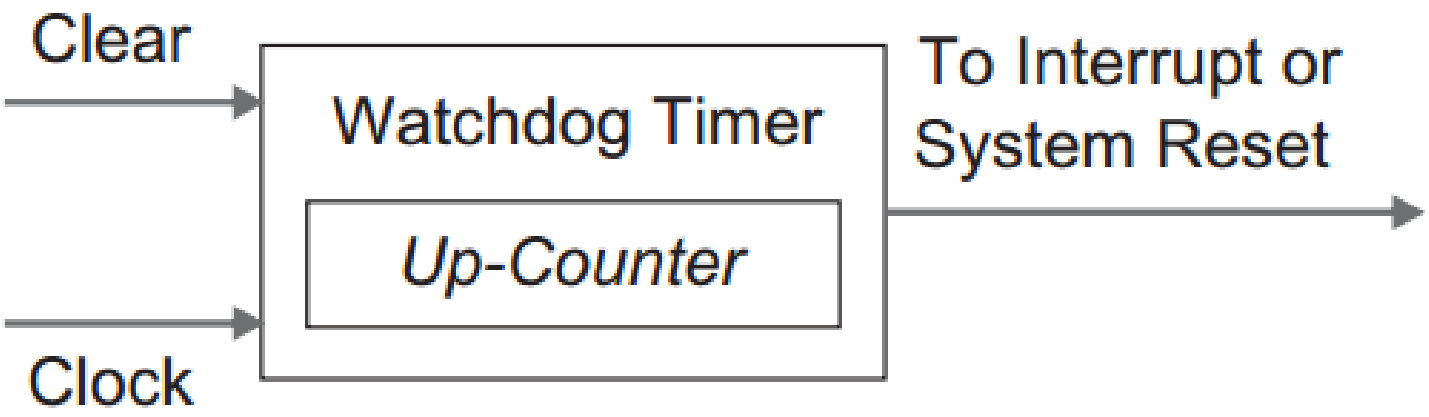


esign

Watchdog Timer

Is used in embedded systems to ensure that:

- certain devices are services at regular intervals;
- software tasks execute according to their prescribed rate;
- CPU functions normally

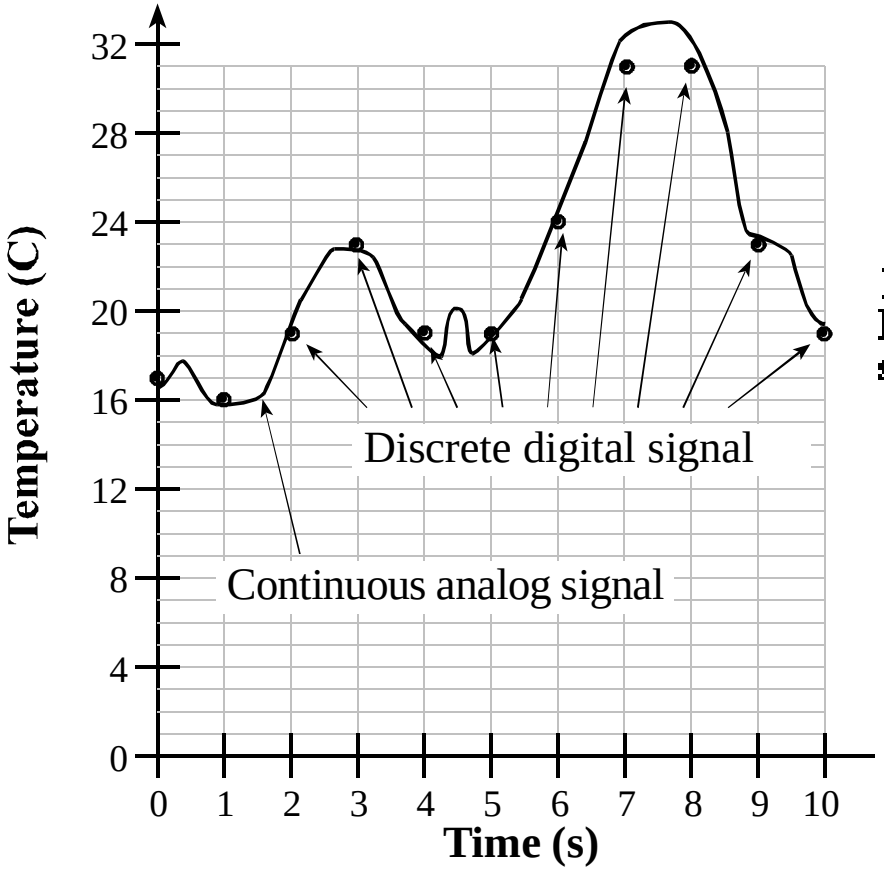


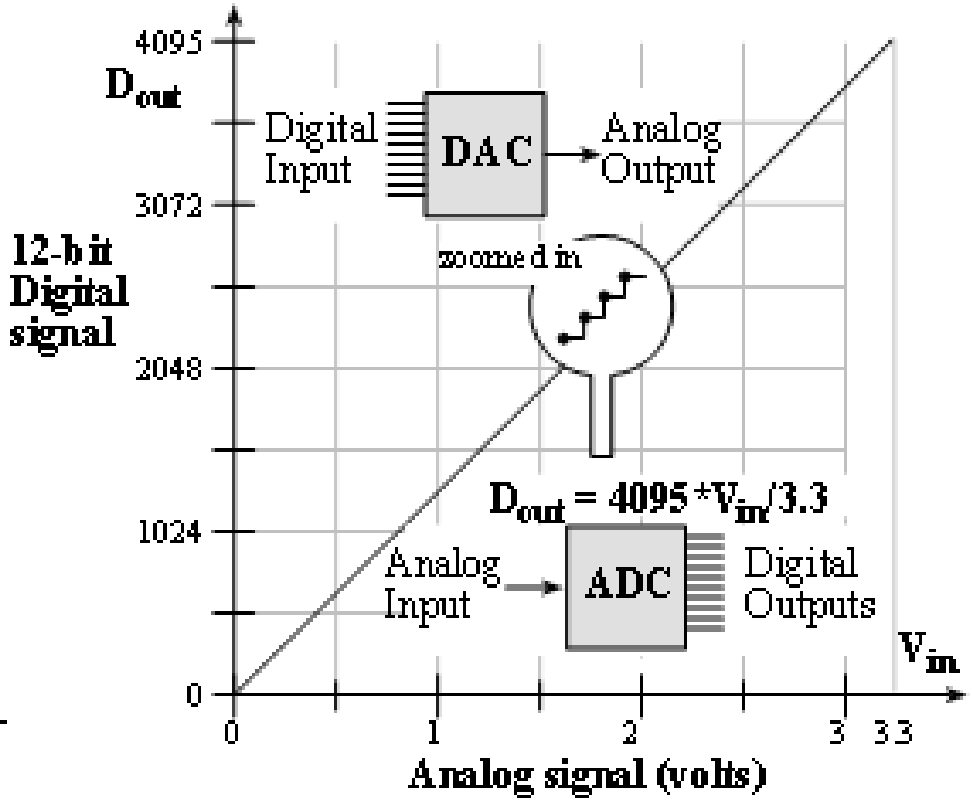
A watchdog timer. Software issues a reset signal via memory-mapped or programmed I/O to reset the timer before it can overflow, issuing a watchdog timer interrupt.

A/D Converters

- Analog-to-digital conversion converts continuous (analog) signals from various transducers and devices into discrete (digital) ones.
- Similar circuitry can be used to convert temperature, sound, pressure, and other inputs from transducers using a variety of sampling schemes to perform the conversion.
- The output of A/D circuitry is a discrete version of the time-varying signal being monitored.
- The discrete version of the continuous value can be treated as a scaled number.
- The key factor in the service of A/D circuitry for time varying signals is the sampling rate (the Nyquist rate).

Analog-to-Digital Converter (ADC)





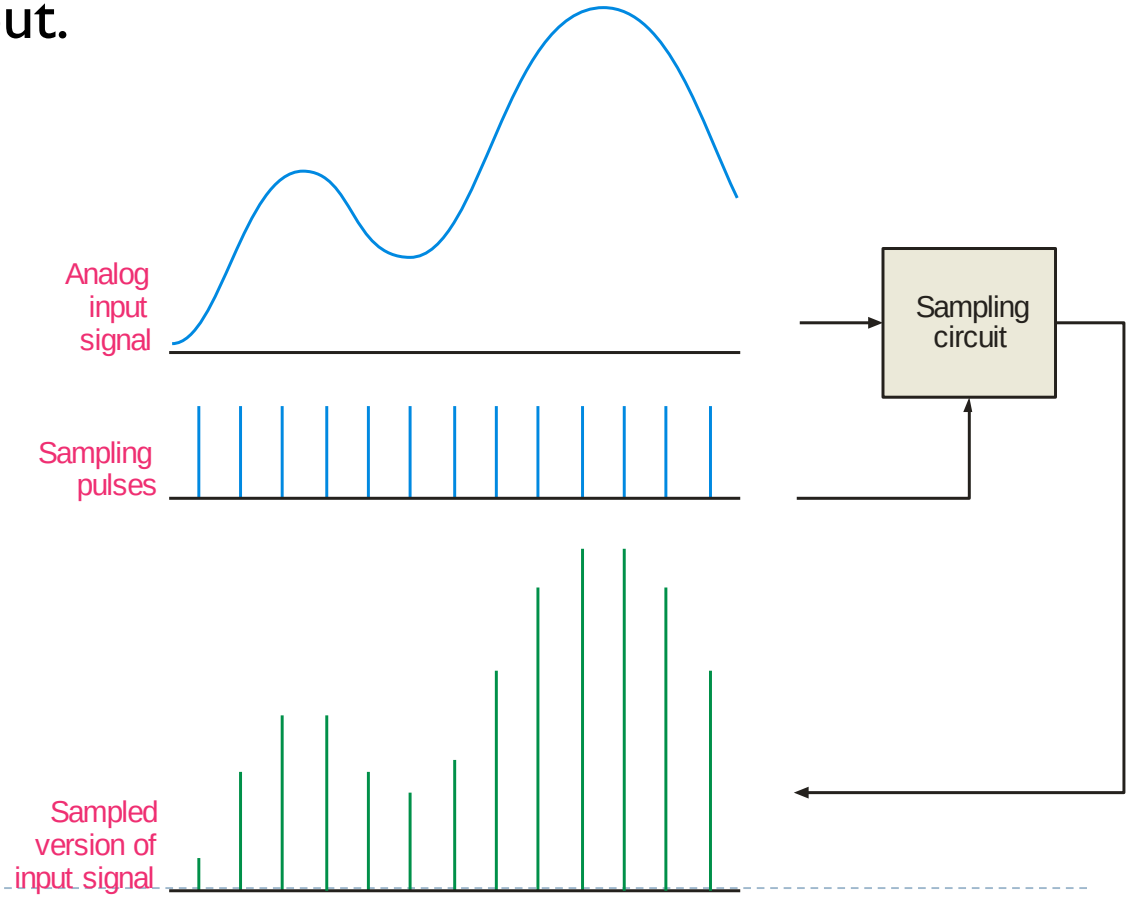
ADC on TM4C123

- Sampling Range/Resolution
  - 3.3V internal reference voltage
  - 0x000 at 0 V input
  - 0xFFF at 3.3 V
  - resolution = range/precision
    = 3.3V/**4096** alternatives < 1mV

Signal Sampling

Most input signals to an electronic system start out as analog signals. For processing, the signal is normally converted to a digital signal by sampling the input.

Before sampling, the analog input must be filtered with a low-pass anti-aliasing filter. The filter eliminates frequencies that exceed a certain limit that is determined by the sampling rate.



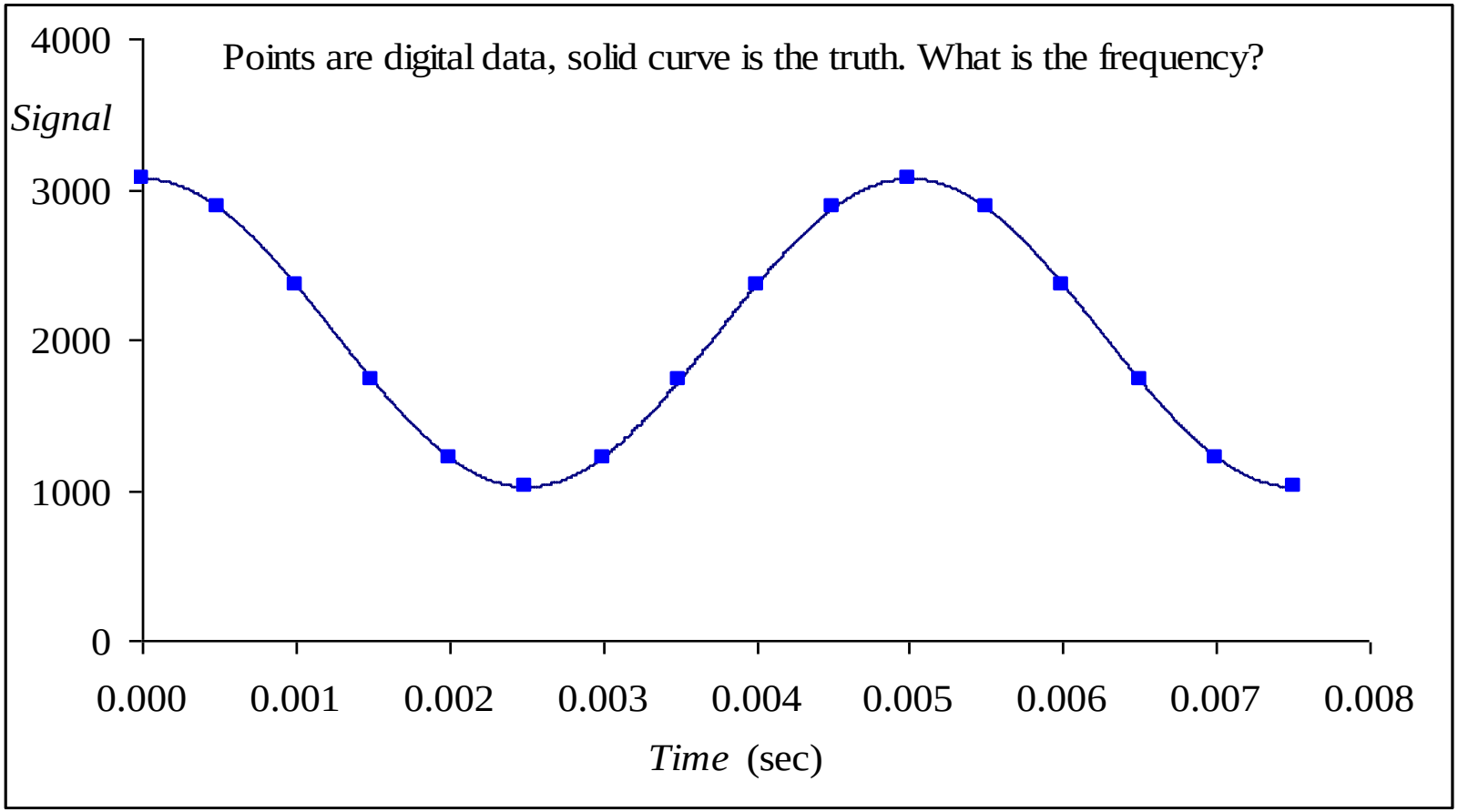
Nyquist Theorem

- A *bandlimited* analog signal that has been sampled can be perfectly reconstructed from an infinite sequence of samples if the sampling rate f_s exceeds $2f_{max}$ samples per second, where f_{max} is the highest frequency in the original signal.
  - If the analog signal does contain frequency components larger than $(1/2)f_s$, then there will be an **aliasing** error.
  - Aliasing is when the digital signal appears to have a different frequency than the original analog signal.

- **Valvano Postulate**: If f_{max} is the largest frequency component of the analog signal, then you must sample more than ten times f_{max} in order for the reconstructed digital samples to look like the original signal when plotted on a voltage versus time graph.

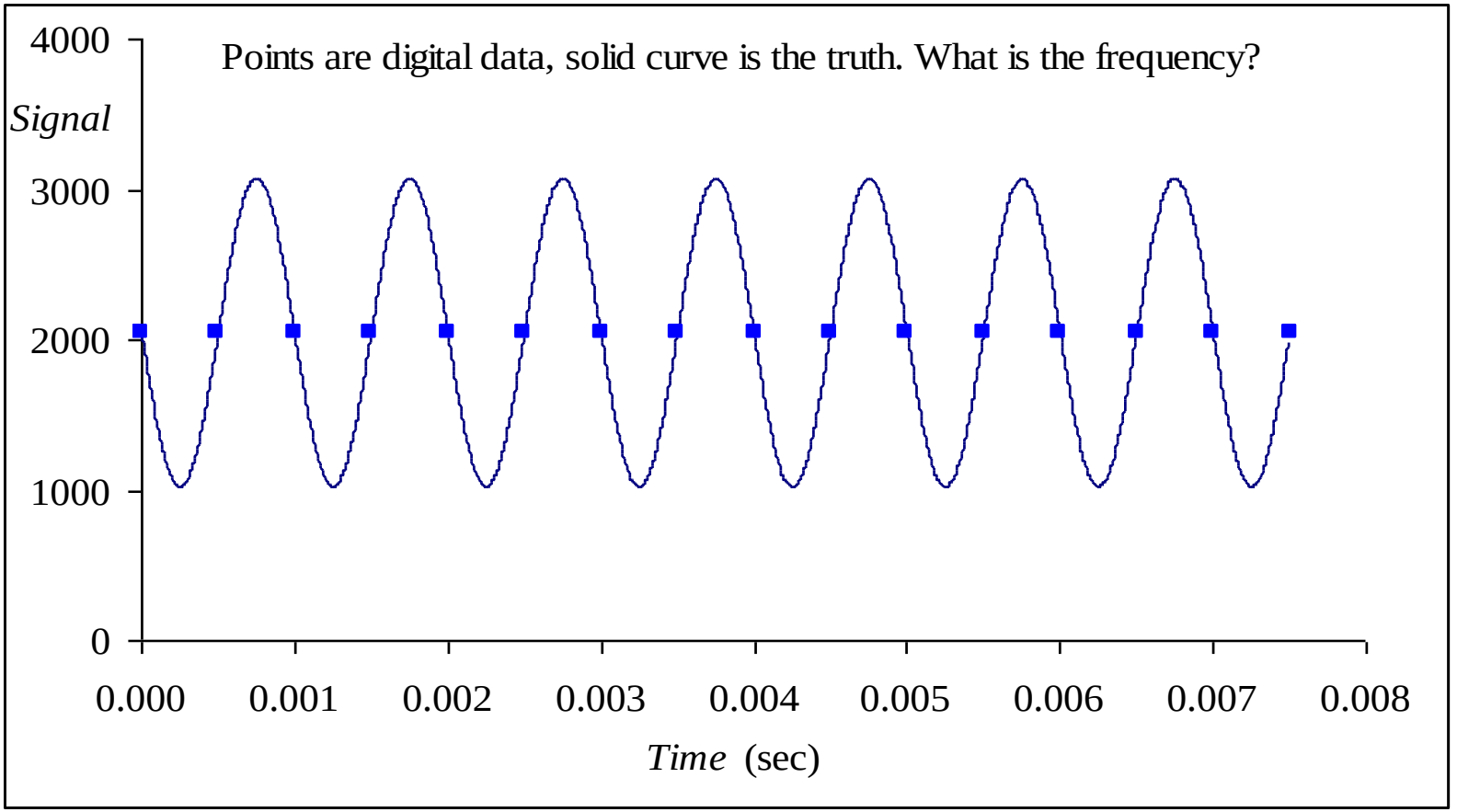
Sampling (option 1)

- 200Hz signal sampled at 2000Hz



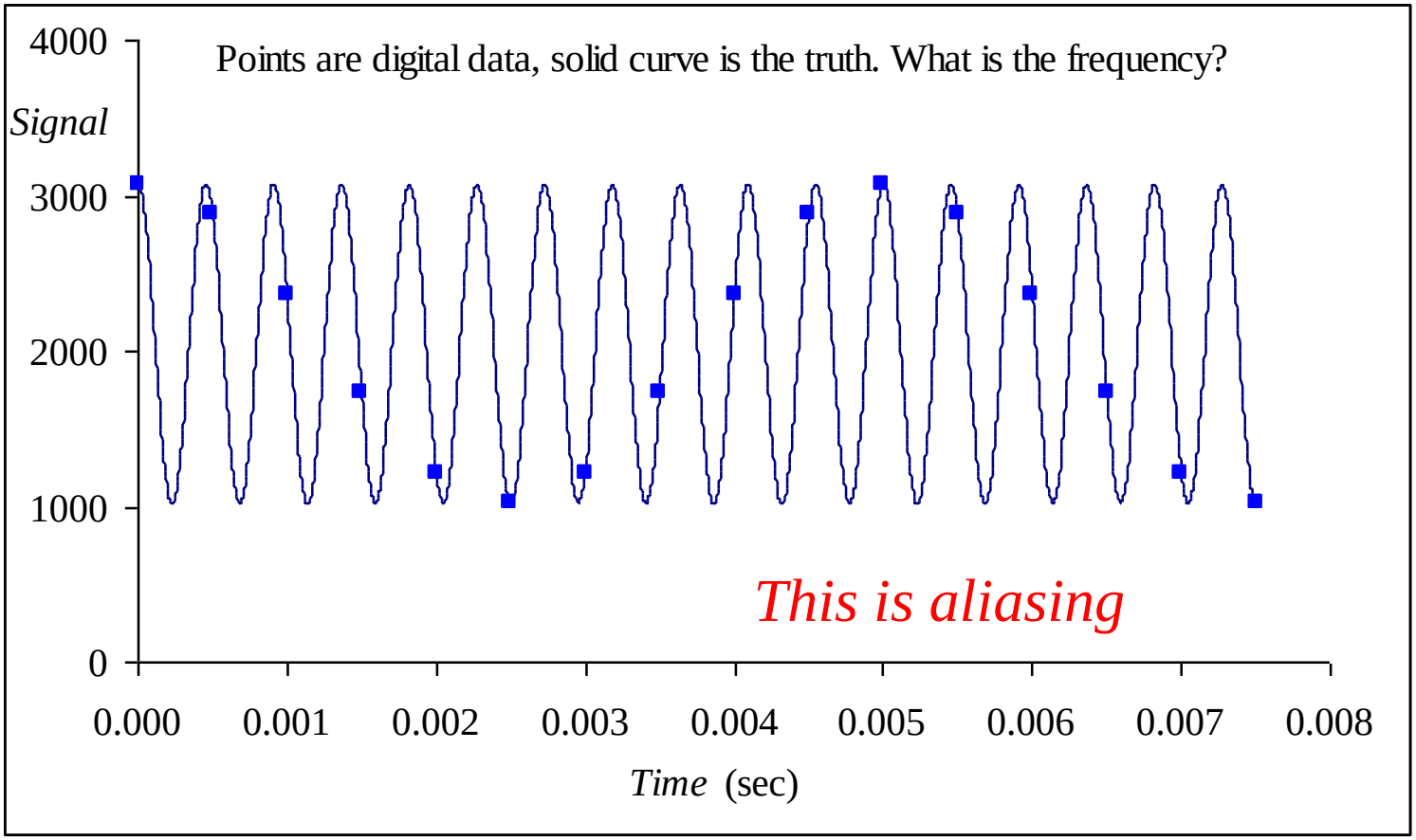
Sampling (option 1)

- 1000Hz signal sampled at 2000Hz



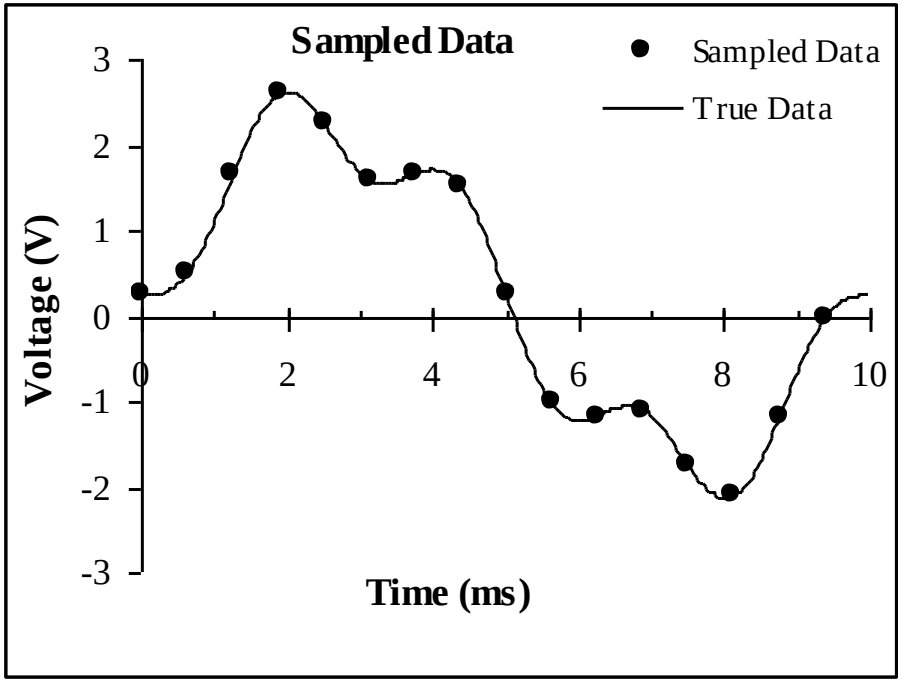
Sampling (option 1)

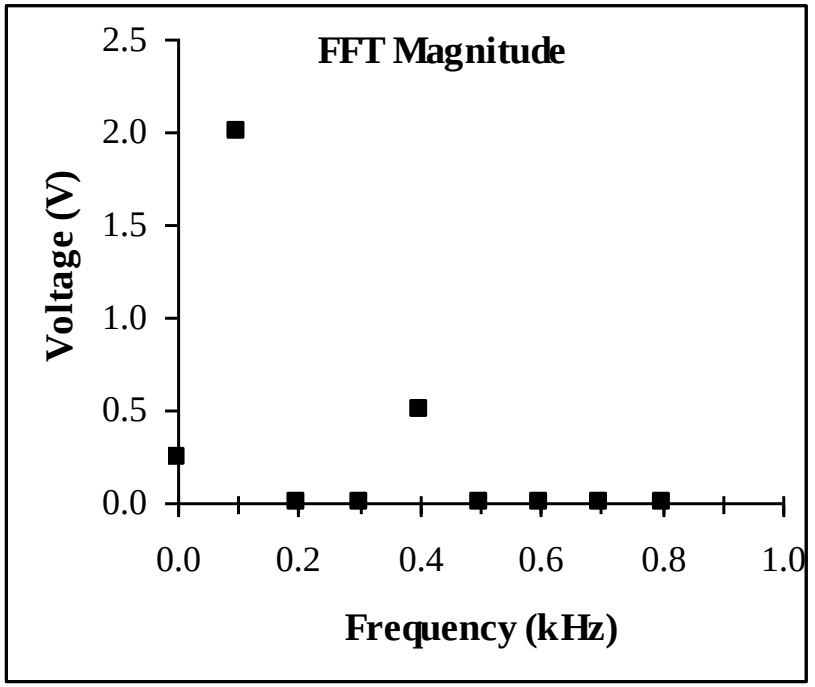
- 2200Hz signal sampled at 2000Hz



Sampling (option 2)

- A signal with DC, 100Hz and 400Hz sampled at 1600Hz

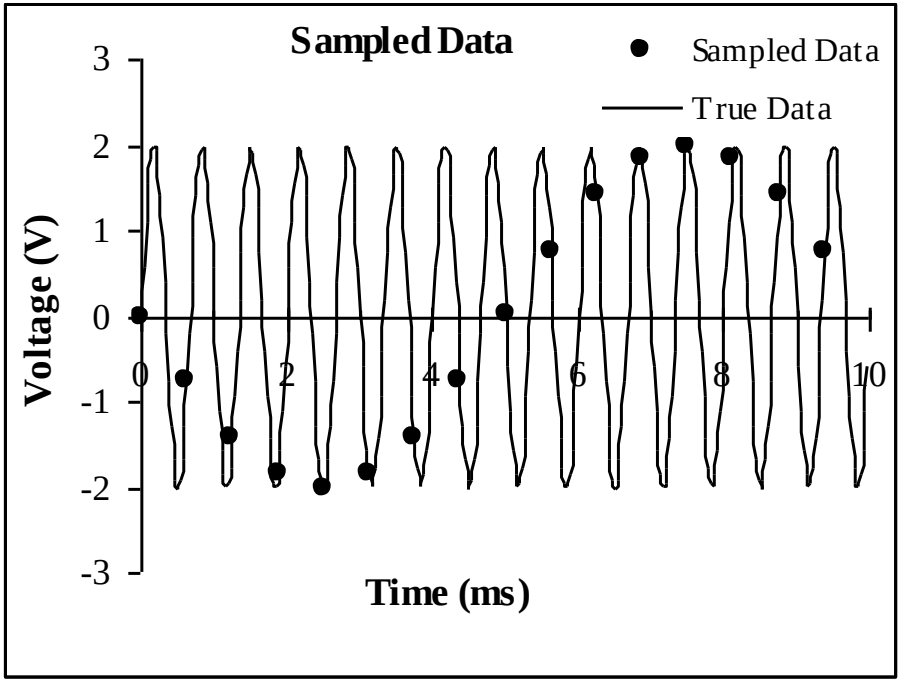


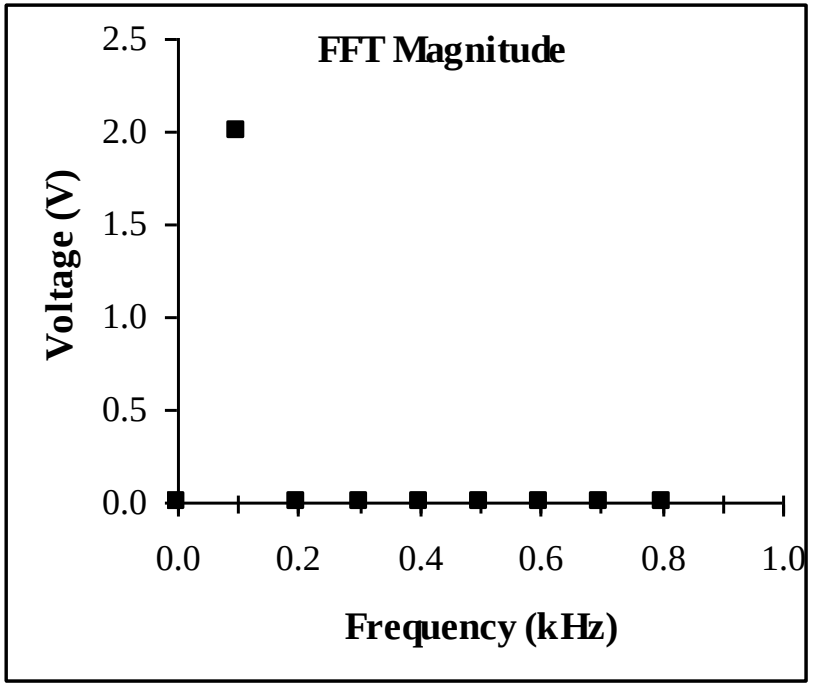


Sampling (option 2)

- 1500Hz signal sampled at 1600Hz

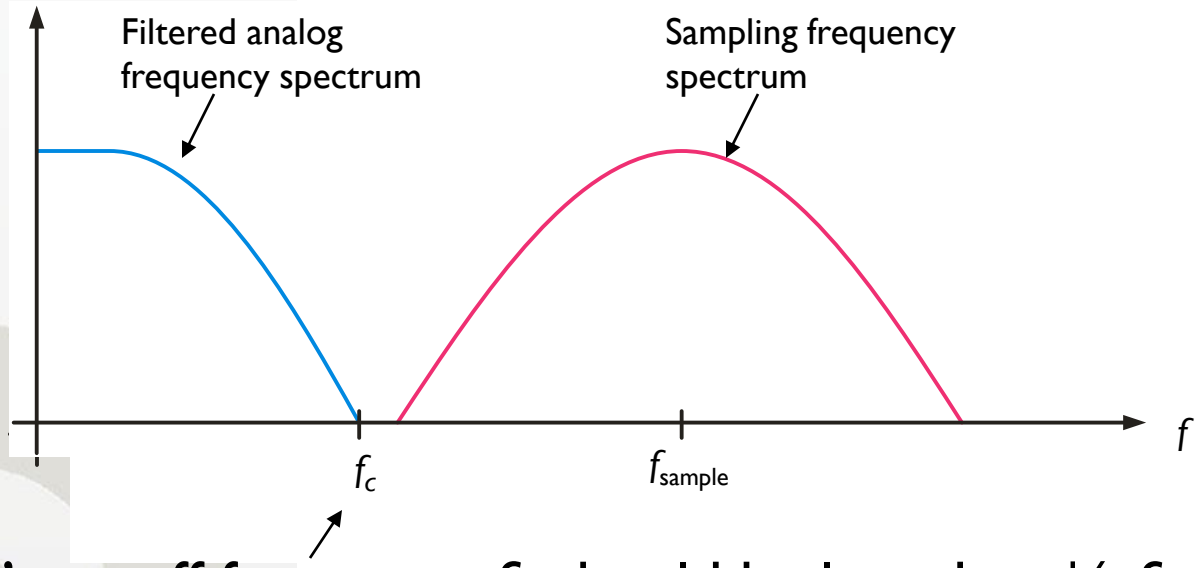
This is aliasing





Anti-Aliasing Filter

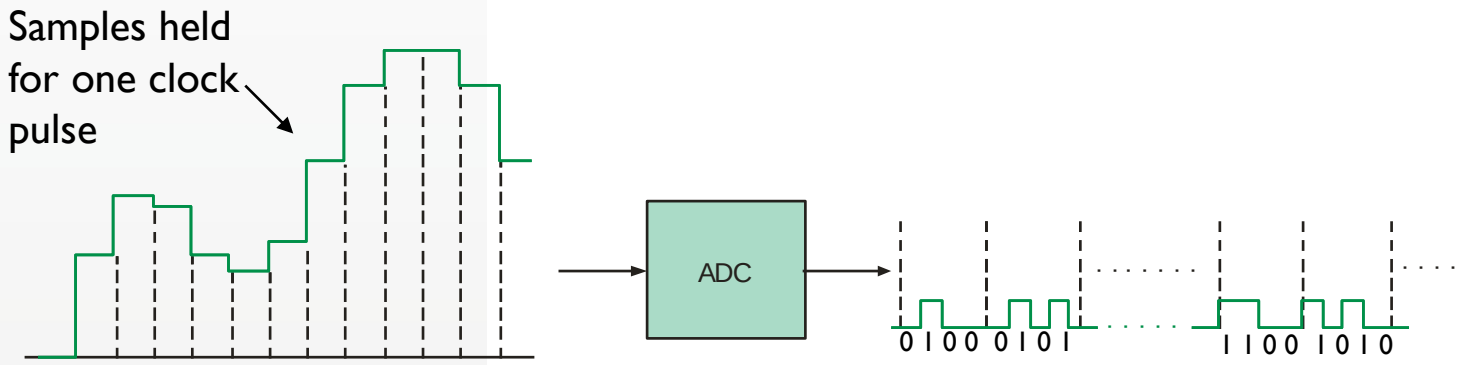
The anti-aliasing filter is a low-pass filter that limits high frequencies in the input signal to only those that meet the requirements of the sampling theorem.



The filter's cutoff frequency, f_c, should be less than ½ f_{sample}.

A/D Conversion

Following the anti-aliasing filter, is the sample-and-hold circuit and the analog-to-digital converter. At this point, the original analog signal has been converted to a digital signal.



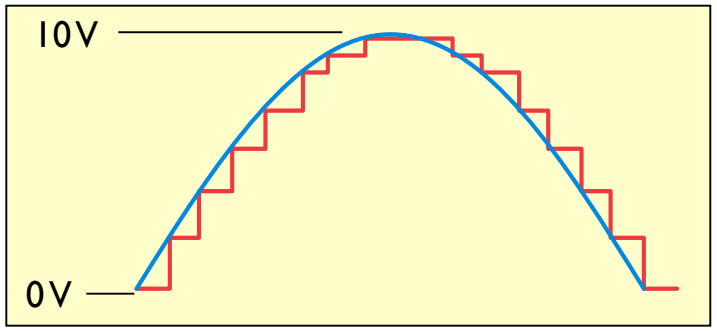
Many ICs can perform both functions on a single chip and include two or more channels.

A/D Conversion

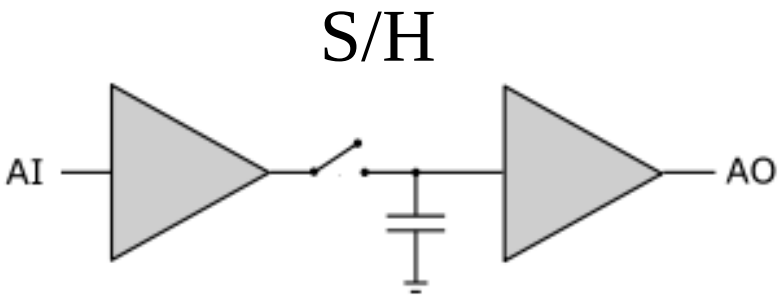
To process naturally occurring analog quantities with a digital system, the analog signal is converted to digital form after the anti-aliasing filter.

The first step in converting a signal to digital form is to use a sample-and-hold circuit. This circuit samples the input signal at a rate determined by a clock signal and holds the level on a capacitor until the next clock pulse.

A positive half-wave from 0-10V is shown in blue. The sample-and-hold circuit produces the staircase representation shown in red.



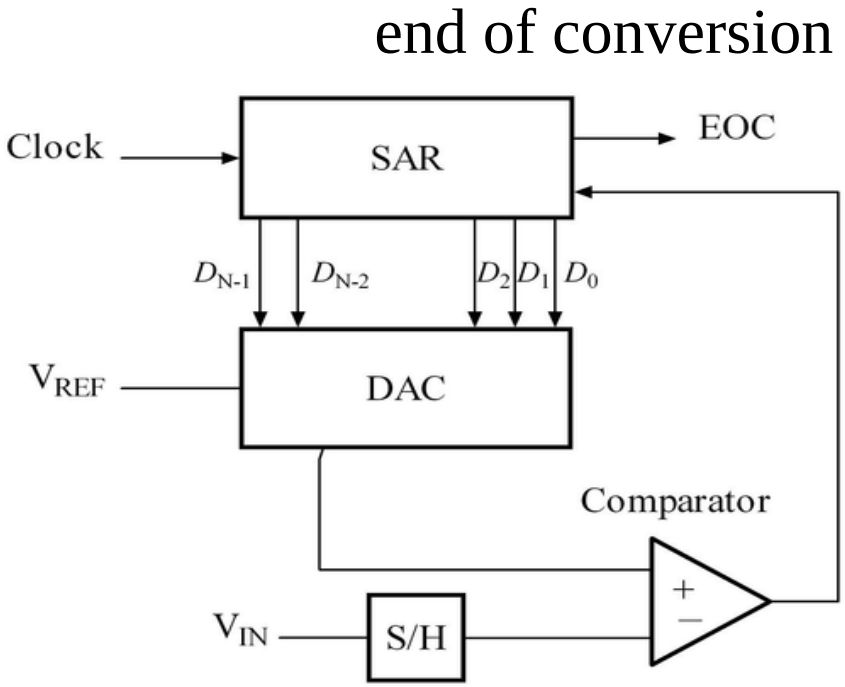
Sample-And-Hold Circuit



- **Analog Input (AI) is sampled when the switch is closed and its value is *held* on the capacitor where it becomes the Analog Output (AO)**

Analog-to-Digital Converter (ADC)

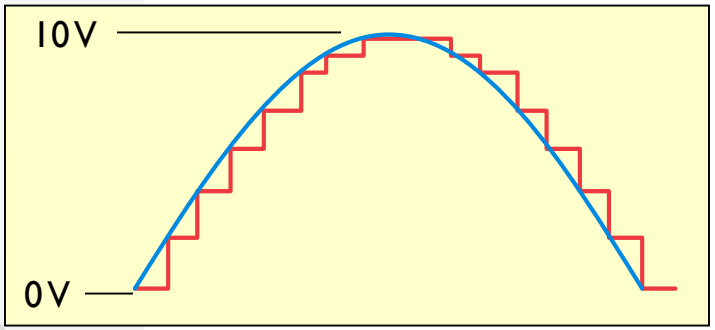
- Successive approximation ADC
  - V_{IN} is approximated as a static value in a *sample and hold* (S/H) circuit
  - the *successive approximation register* (SAR) is a counter that increments each *clock* as long as it is enabled by the *comparator*
  - the output of the SAR is fed to a DAC that generates a voltage for comparison with V_{IN}
  - when the output of the DAC = V_{IN} the value of SAR is the digital representation of V_{IN}



A/D Conversion

Example What is the maximum unsigned binary value for the waveform?

Solution $10\,V = 1010_2\,V$. The table lists the quantized binary values for all of the steps.



Peak = 10 V

0.0000
10.0001
100.0001
101.1110
111.0111
1000.1011
1001.1001
1010.0000
1010.0000
1001.1001
1000.1011
111.0111
101.1110
100.0001
10.0001
0.0000

D/A Converters

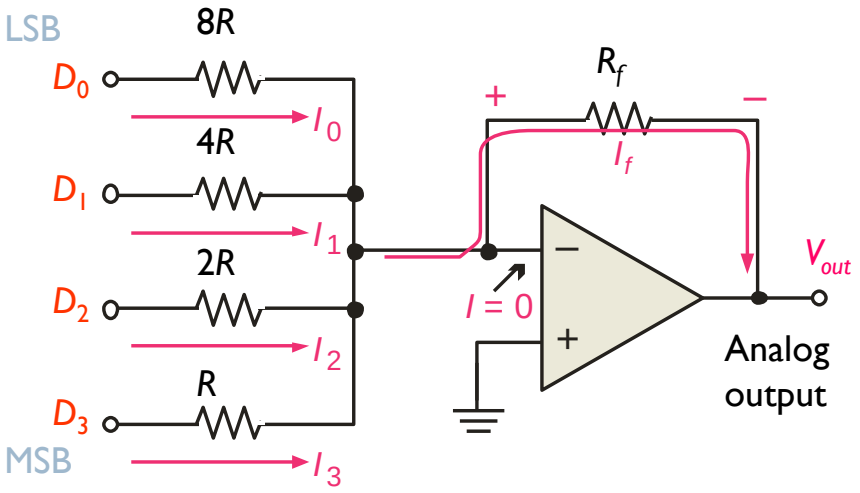
- Digital-to-analog conversion performs the inverse function of A/D circuitry.
- Converts a discrete quantity to a continuous one.
- D/A devices are used to allow the computer to output analog voltages based on the digital version stored internally.
- Communication with D/A circuitry uses one of the three input/output methods discussed.

D/A Converters

Binary-weighted-input DAC:

The binary-weighted-input DAC is a basic DAC in which the input current in each resistor is proportional to the column weight in the binary numbering system. It requires very accurate resistors and identical HIGH level voltages for accuracy.

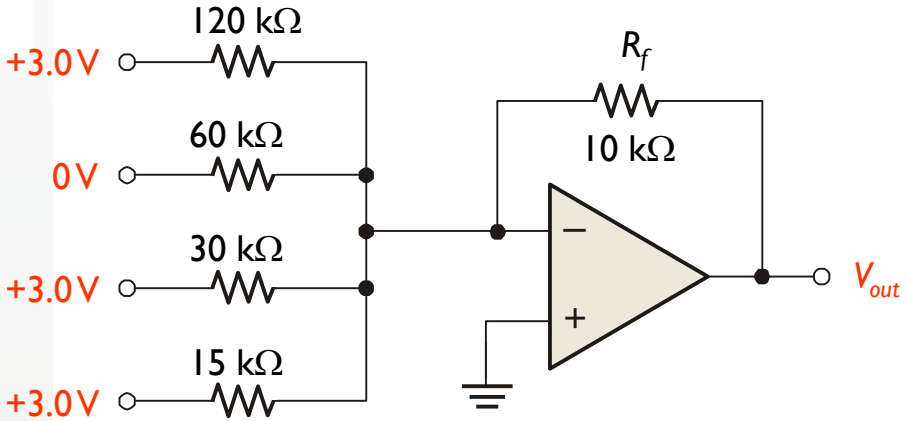
The MSB is represented by the largest current, so it has the smallest resistor. To simplify analysis, assume all current goes through R_f and none into the op-amp.



D/A Converters

Binary-weighted-input DAC:

Example A certain binary-weighted-input DAC has a binary input of 1101. If a HIGH = +3.0 V and a LOW = 0 V, what is V_{out}?



Solution

$$I_{out} = -(I_0 + I_1 + I_2 + I_3)$$

$$= -\left(\frac{3.0\text{ V}}{120\text{ k}\Omega} + 0\text{ V} + \frac{3.0\text{ V}}{30\text{ k}\Omega} + \frac{3.0\text{ V}}{15\text{ k}\Omega}\right) = -0.325\text{ mA}$$

$V_{out} = I_{out} R_f = (-0.325\text{ mA})(10\text{ k}\Omega) = -3.25\text{ V}$

D/A Converters

R-2R ladder:

The *R-2R* ladder requires only two values of resistors. By calculating a Thevenin equivalent circuit for each input, you can show that the output is proportional to the binary weight of inputs that are HIGH.

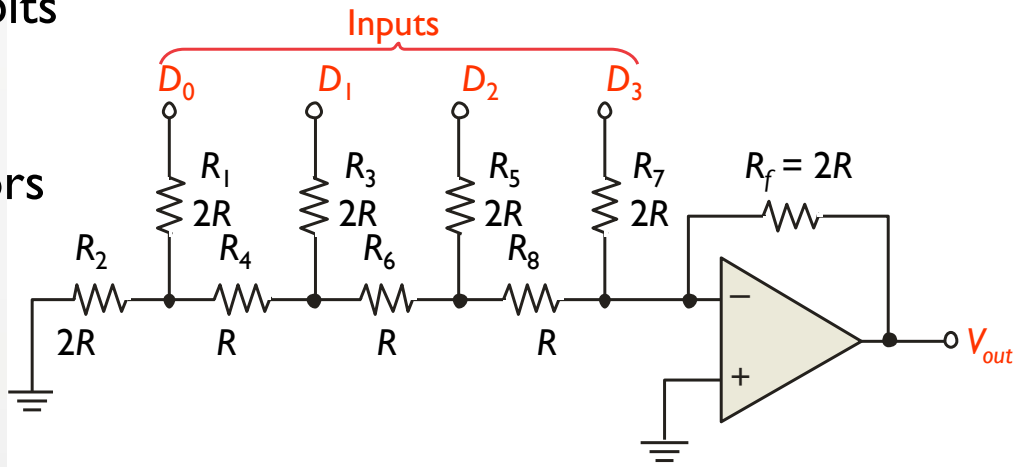
Each input that is HIGH contributes to the output: $V_{out} = -\frac{V_S}{2^{n-i}}$

where V_S = input HIGH level voltage

n = number of bits

i = bit number

For accuracy, the resistors must be precise ratios, which is easily done in integrated circuits.

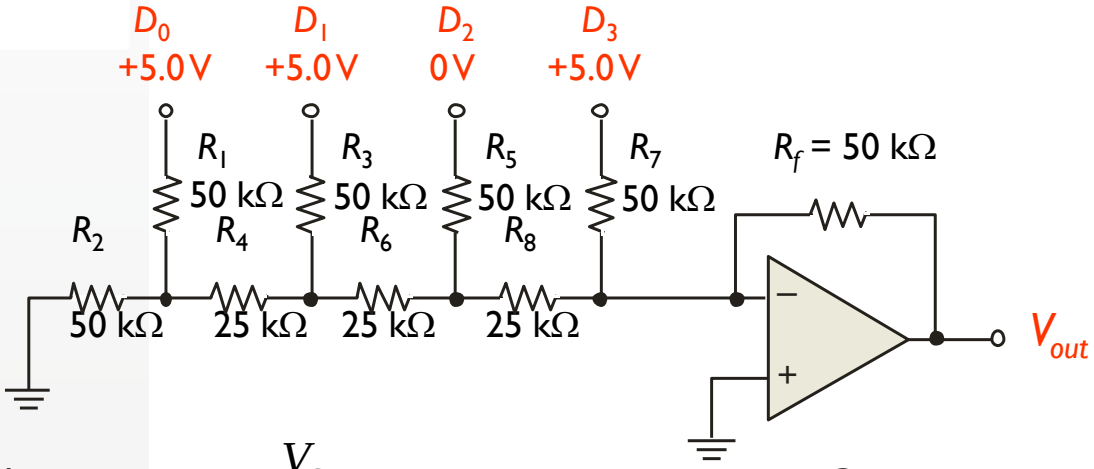


D/A Converters

R-2R ladder:

Example

An *R-2R* ladder has a binary input of 1011. If a HIGH = +5.0 V and a LOW = 0 V, what is V_{out}?



Solution

Apply $V_{out} = -\frac{V_S}{2^{n-i}}$ to all inputs that are HIGH, then sum the results.

$$V_{out}(D_0) = -\frac{5\text{ V}}{2^{4-0}} = -0.3125\text{ V} \qquad V_{out}(D_1) = -\frac{5\text{ V}}{2^{4-1}} = -0.625\text{ V}$$

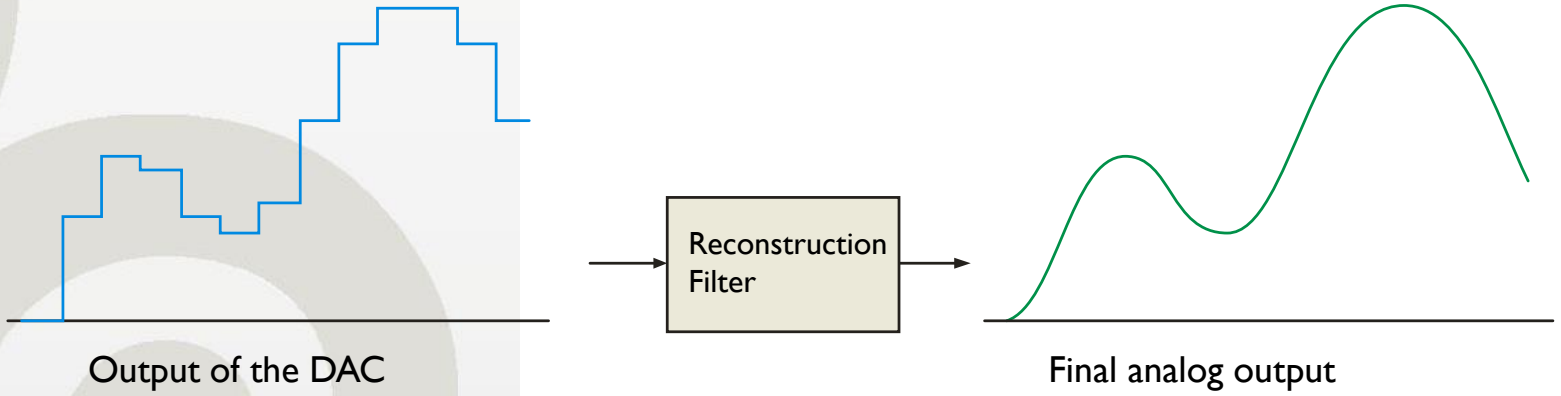
$$V_{out}(D_3) = -\frac{5\text{ V}}{2^{4-3}} = -2.5\text{ V}$$

Applying superposition, V_{out} = −3.43 V

D/A Converters

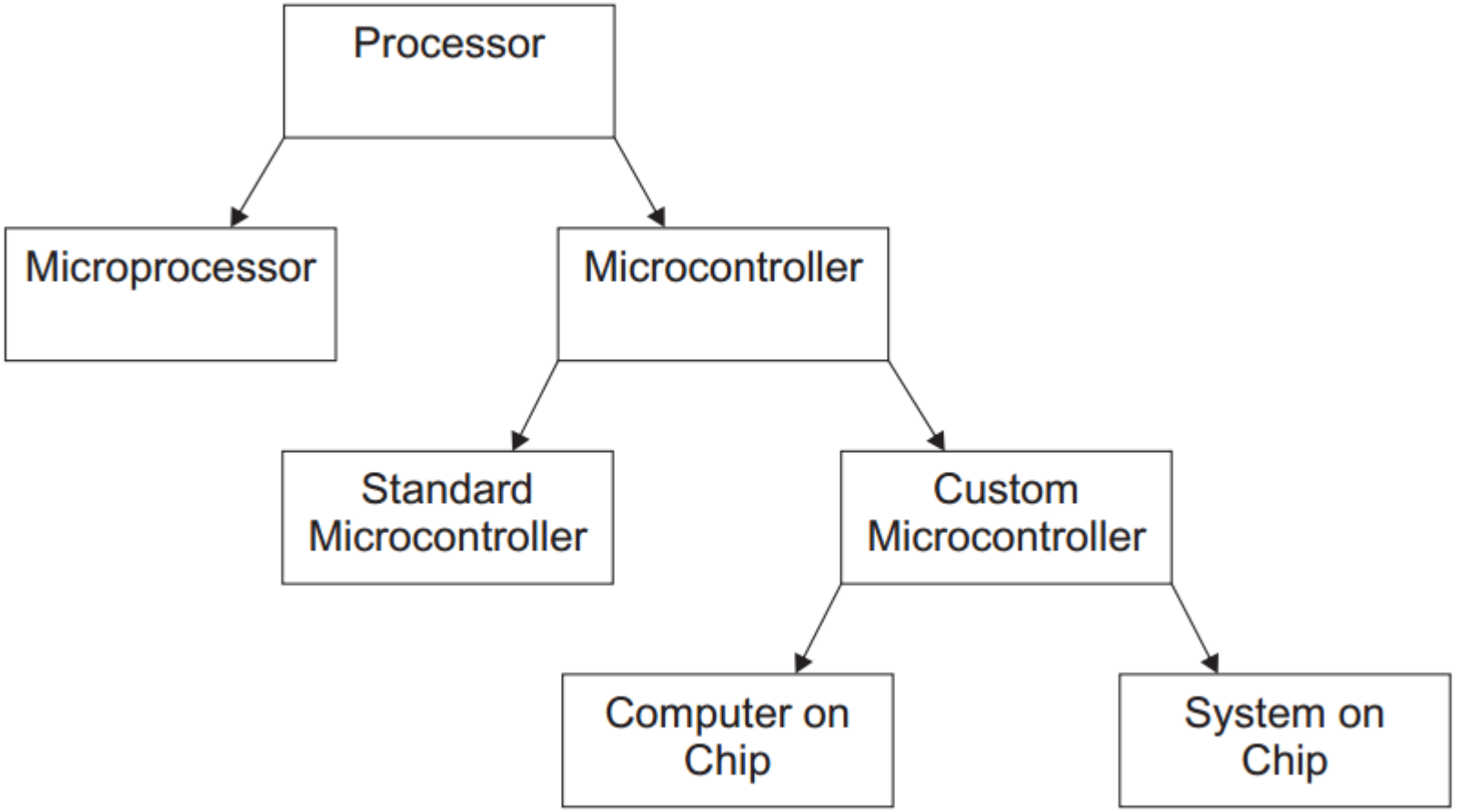
Reconstruction Filter:

After converting a digital signal to analog, it is passed through a low-pass "reconstruction filter" to smooth the stair steps in the output. The cutoff frequency of the reconstruction filter is often set to the same limit as the anti-aliasing filter, to block higher harmonics due to the digitizing process.



Processor Technology

Principal evolution paths of processor technologies



Standard Microcontroller

Modern microcontroller could contain the following:

- EEPROM or Flash
- SRAM
- ADC with a multiplexer
- DMA controller
- Parallel inputs and outputs
- Serial interface
- Timers and counters
- PWM
- Watchdog timers

Custom Microcontrollers

- Applications specific integrated circuit – a special purpose integrated circuit designed for one application only.
- In essence, these devices are systems on a chip that can include a microprocessor, memory, I/O devices and other specialized circuitry.
- ASICs are used in many embedded applications including image processing, avionics systems, medical systems.
- Real-time design issues are the same for them as they are for most other systems.

FPGAs

- Field programmable gate array (FPGA) – allows construction of a system on a chip with an integrated processor, memory, and I/O.
- It is reprogrammable, even while embedded in the system.
- A reconfigurable architecture allows for the programmed interconnection and functionality of a conventional processor
- Algorithms and functionality are moved from residing in the software side into the hardware side.
- Widely used in embedded, mission-critical systems where fault-tolerance and adaptive functionality is essential.

Any Questions?