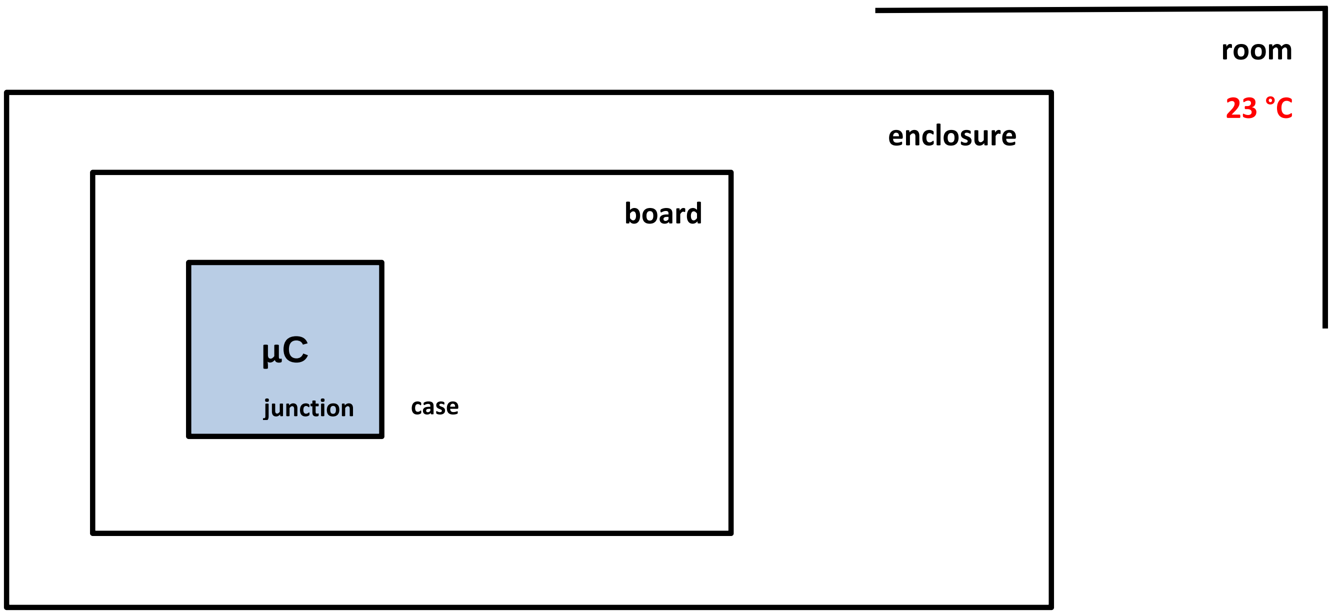
System-Related Design Issues

This document covers design issues that relate more to the overall system, i.e., "what's happening outside of the processor". Some of them may not be so obvious, but they still need to be considered in the design of a real-time embedded system.

Temperature Scenarios

Recall that the operating temperature range of the Piccolo processor *at the semiconductor junction* is −40 to +105 °C. The designer should take care that the actual temperature at the junction never goes outside this range. Below are some scenarios to illustrate the design considerations.

1. **Heat-Rise Budget** – Consider a µC on a board in an enclosure (e.g. cabinet containing a bunch of racks) in an air-conditioned room.



Even if the junction temperature is within the operating range when the room is at "normal room temperature", the worst-case scenarios should be considered…

2. **Air-conditioner fails on hot summer day in hot climate**

   For example, room goes up to 50 °C. So add 27 °C to each of the original values:

room	50 °C
enclosure	
board	
case	
junction	

3. **Heater fails on cold winter day in cold climate**

   For example, room goes down to −45 °C. So subtract 68 °C from each of the original values:

room	−45 °C
enclosure	
board	
case	
junction	

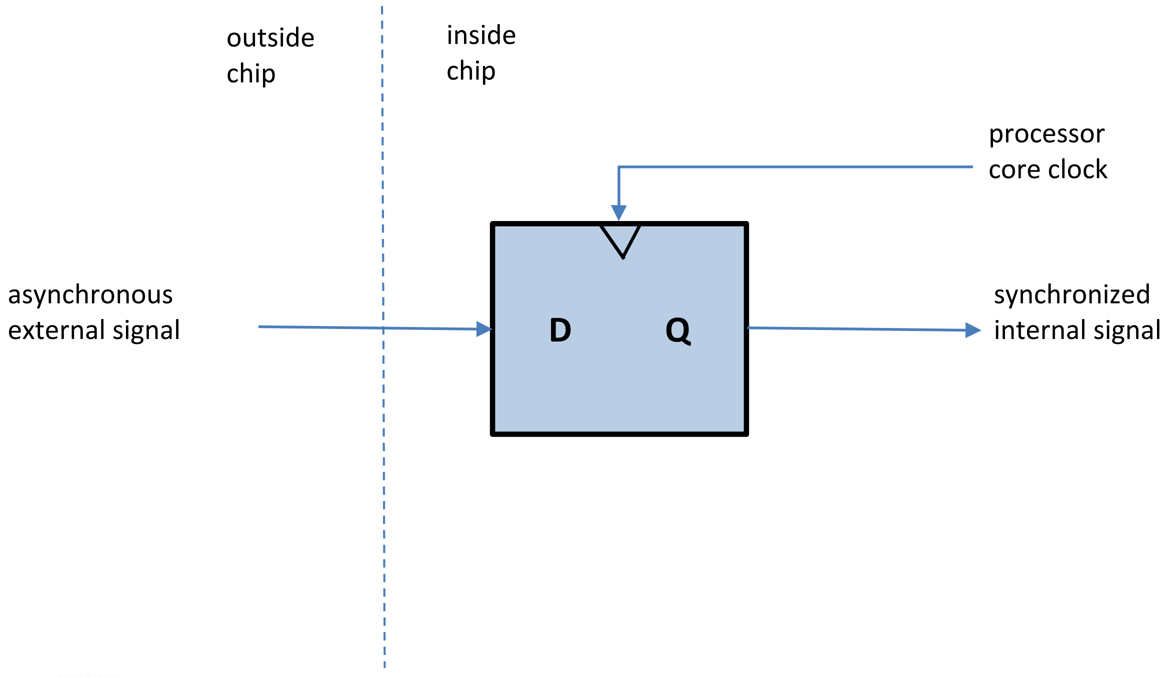
4. **Heater and electronics power fails on cold winter day in cold climate**

   Immediately after power is restored and µC turns on:

room	−45 °C
enclosure	
board	
case	
junction	

Digital Input Integrity – Metastability

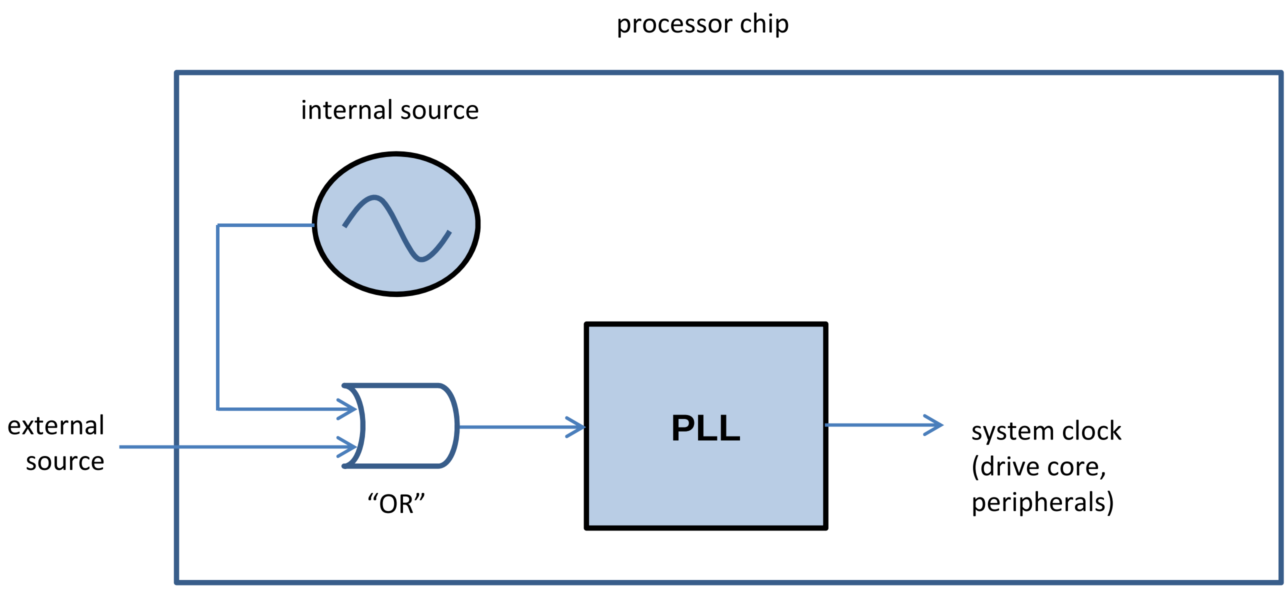
Digital signals from the outside world that are inputs to a processor need to be synchronized to the processor's core clock so that their state is known at the clock edge. This can be done with a D-Q flip-flop in the processor chip, where the clock is the core clock. All flip-flops have a window of time between the set-up and hold times in which the data input D is assumed to not change. However, external signals can change at any time. If an external signal changes within the set-up/hold window, the Q output state is not guaranteed to be known until it "settles". In fact, it can oscillate. The settling time can be infinite in theory. This is known as "metastability". Modern flip-flops have shorter and shorter, but nevertheless, finite windows. So there is a probability that, eventually, an external signal can switch states during the window and cause metastability. If this can cause a deleterious effect to the system, it needs to be mitigated by the hardware and/or firmware design. With good hardware design, the chance of a metastable event can be reduced to a very small probability, but not zero. With good firmware design, the effect of a metastable event can be minimized.



Adobe Acrobat Document

<u>Clock Source</u>

Typically, modern processors have an on-board phase-locked loop (PLL) that generates the system (core) clock from a lower-frequency input clock. The input clock can be from within the chip, i.e., on-chip, or an external source.



Why might the designer want to use an external clock source (rather than the internal clock source) to drive the on-chip PLL?

➔ ***To achieve better performance in one or more of the following areas:***

- frequency accuracy (initial frequency accuracy)
- frequency stability
  - variation in frequency over operating temperature range
  - variation in frequency over time (aging)
- timing jitter
- phase noise

look at oscillator specs

Timing jitter:

Phase noise:

Examples that need accurate timing:

→ clock (to keep time)

→ telecom signalling (digital data)

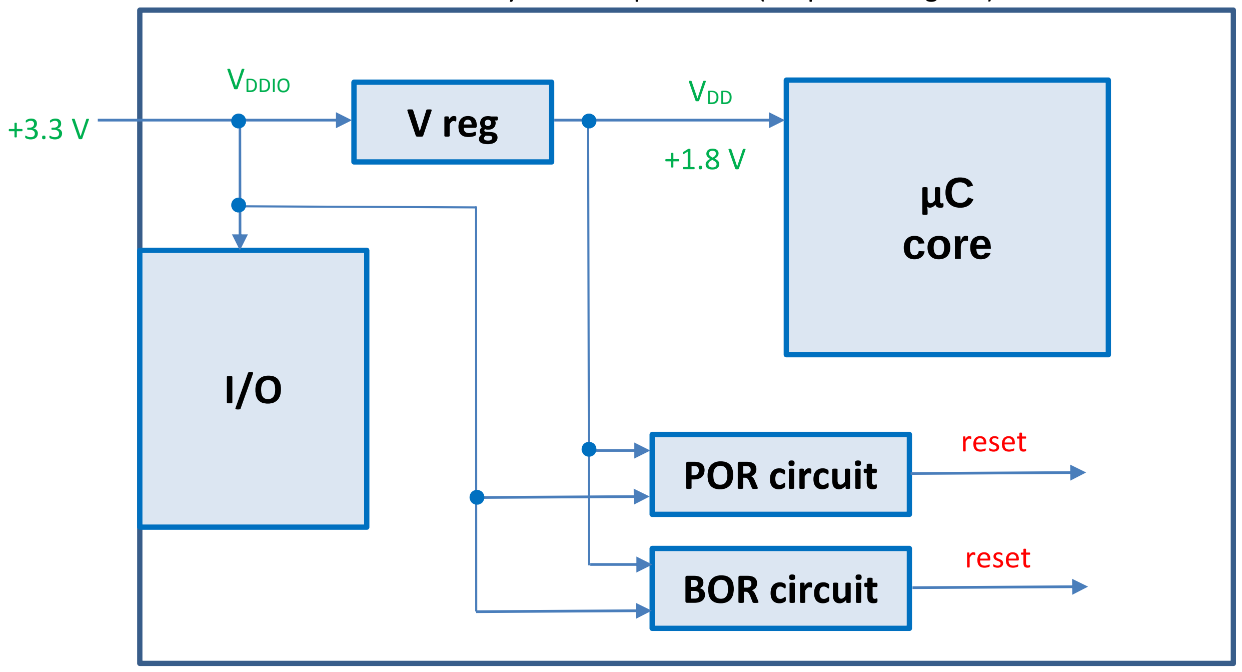
Can you think of any other reasons to use an external source?

Power-On Reset (POR) and Brown-Out Reset (BOR) Features

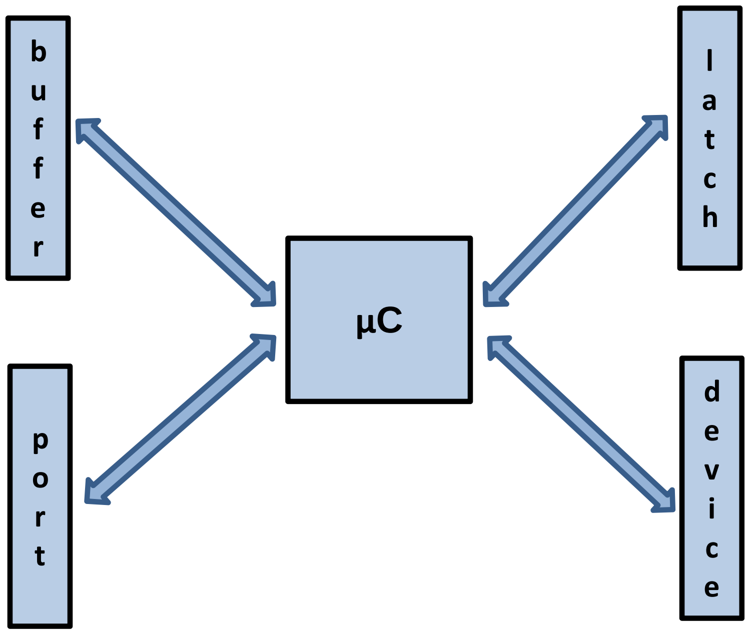
POR:
– resets processor when rising supply voltage (i.e., power-up) is detected

BOR:
– resets processor when sagging supply voltage (i.e., brown-out) is detected

POR and BOR circuitry in c2000 processor (simplified diagram)



1. The reset signal from the POR can be sent to external hardware so that the external hardware can come up to a known defined state *before* the µC CPU begins execution. That way, the µC can start interfacing, i.e., talking, to the external hardware in a deterministic way.



2. The reset signal from the BOR can be used to trigger a subroutine, e.g. non-maskable interrupt (NMI) function, to finish any critical activities that must be completed.

   e.g. to finish the write-cycle sequence of data into an EEPROM