ID A11C: Hardware Design Fundamentals for MCU-based Embedded Systems
Mr. Mitch Ferguson

- Applications Engineer Manager
- Specializes support design teams develop low-noise systems using MCUs.
- Over 15 years of system-level design experience
- Over 7 years of experience as an application engineer.
- As a hardware engineer and engineering manager, he has been involved in design in power distribution controls, automotive and fire alarm systems with focus on EMI/EMS issues.
- Bachelor of science in electrical engineering from Cleveland State University
In the session 110C, Renesas Next Generation Microcontroller and Microprocessor Technology Roadmap, Ritesh Tyagi introduces this high level image of where the Renesas Products fit. The big picture.
This is where our session, A11C Hardware Design Fundamentals, is focused within the ‘Big picture of Renesas Products’, Microcontroller and Microprocessors.
Microcontroller and Microprocessor Line-up

32-bit

- Superscalar, MMU, Multimedia
  - Up to 1200 DMIPS, 45, 65 & 90nm process
  - Video and audio processing on Linux
  - Server, Industrial & Automotive

- High Performance CPU, Low Power
  - Up to 500 DMIPS, 150 & 90nm process
  - 600uA/MHz, 1.5 uA standby
  - Medical, Automotive & Industrial

- High Performance CPU, FPU, DSC
  - Up to 165 DMIPS, 90nm process
  - 500uA/MHz, 2.5 uA standby
  - Ethernet, CAN, USB, Motor Control, TFT Display

- Legacy Cores
  - Next-generation migration to RX

8-bit

- General Purpose
  - Up to 10 DMIPS, 130nm process
  - 350 uA/MHz, 1uA standby
  - Capacitive touch

- Ultra Low Power
  - Up to 25 DMIPS, 150nm process
  - 190 uA/MHz, 0.3uA standby
  - Application-specific integration

- Embedded Security
  - Up to 25 DMIPS, 180, 90nm process
  - 1mA/MHz, 100uA standby
  - Crypto engine, Hardware security
Microcontroller and Microprocessor Line-up

Superscalar, MMU, Multimedia
- Up to 1200 DMIPS, 45, 65 & 90nm process
- Video and audio processing
- Server, Industrial & Automation

High Performance CPU, Low Power
- Up to 1200 DMIPS, 45, 65 & 90nm process
- Ultra Low Power
- Medical, Automotive & Industrial

General Purpose
- Up to 10 DMIPS, 130nm process
- 350 uA/MHz, 1uA standby
- Capacitive touch

Low Power
- Up to 100 DMIPS, 150nm process
- 1uA/MHz, 0.3uA standby
- Application-specific integration

Embedded Security
- Up to 25 DMIPS, 180, 90nm process
- 1mA/MHz, 100uA standby
- Crypto engine, Hardware security

All of Them!
One of the most significant innovations that has changed the embedded world is the tremendous amount of peripheral integration into a single MCU device. Analog and digital circuits that used to be external devices are not being pulled into one chip, eliminating the need for complex external bus systems, translators and glue logic. It’s almost like you can just drop the chip down and start writing software.
Integration has made hardware easier but not something that can be ignored

Though integration has made the task of hardware design much easier there are still many fundamental hardware decisions that have to be made for each design. One thing that has not changed about making a hardware mistake, you can’t always easily fix it in code so it is worth understanding the design challenges and making sure the right decision is made.
During this presentation we will look at some of the basic decisions which need to be made when designing an MCU system. One of the fundamental decisions which has to be made is the clock selection. We will then look at POR and LVD. After discussing POR and LVD we will look at WDT circuit expectations. Finally we will look at some common situation when connecting input and output circuits
How much time do you spend designing hardware

1. Firmware Only
2. Both Hardware and Firmware
3. Hardware Engineer
4. Architecture Level only
Clock Circuit Selection

- Clocking circuit criteria
  - Startup time
  - Accuracy
  - Reliability
  - Cost

- Clock alternatives
  - Crystal
  - Ceramic Resonator
  - On-Chip Oscillator
  - Compensated External Oscillator (TXO)

What are the important considerations when selecting a clock oscillator (Audience question)

Not all of the criteria we have listed is important in all design situations, however, the one or all of the items listed is usually important in a given design. The other consideration is what are the available clock options. Of the ones listed the compensated external oscillator is not used that often, but does become important in applications involving RF.

External Oscillator typically used for
- Very high accuracy
- Spread Spectrum
- Driving multiple devices
## Clock Comparison – Arranged Best to Worst

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>Cost</th>
<th>Reliability</th>
<th>Startup Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>External</td>
<td>On-Chip</td>
<td>On-Chip</td>
<td>On-Chip (&lt;10 cycles)</td>
</tr>
<tr>
<td>(±25 ppm or better)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crystal</td>
<td>Resonator ($0.16) w/caps</td>
<td>Resonator</td>
<td>Resonator (100 uS)</td>
</tr>
<tr>
<td>(±50-200 ppm)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resonator</td>
<td>Crystal ($0.20)</td>
<td>Crystal/External</td>
<td>Crystal (1-5 mSec)</td>
</tr>
<tr>
<td>(&gt;0.5%)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>On-Chip</td>
<td>External ($2.91)</td>
<td></td>
<td>External (10 mSec)</td>
</tr>
<tr>
<td>(&gt;2%)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

0.5% = 5000 ppm

This chart will compare the available clocking options with respect to the important characteristics we listed. The chart is constructed with the best option at the top and the worst performer at the bottom.

From an accuracy standpoint there is no doubt that the external oscillator has the best characteristics. The crystal oscillator is next, followed by the ceramic resonator and on-chip oscillator (OCO).

The on-chip oscillator has the best cost since it is included as part of the MCU. The resonator is still less expensive than the crystal though it seems the price difference is continuing to decrease. The external oscillator is not cheap but that’s not the reason you would choose it.

Reliability is always subject to some discussion but generally it is accepted that the OCO will be the most reliable since it is designed into the MCU. Though the resonator and crystal circuits will be close the ceramic resonator is usually considered to have the edge compared to the crystal circuits.

From a start-up time perspective the OCO is the clear winner. I have listed the startup time as 10 cycles but in many cases it is faster than this. Ceramic resonators are often rated by their manufacturers to start-up in 100 uS. Though it is possible to start a crystal circuit up faster than a millisecond that is not the typical design. External oscillators often start-up very slowly since they are designed for stability.

Notice that, except for the accuracy, the OCO looks like almost like an ideal choice.
So what is the required accuracy in an application? This chart shows a few key applications and communication methods with their accuracy requirements. The chart also shows where this lines up compared to the different clock alternatives.

Notice that resonators can be used for CAN and USB, these special resonators are “binned” and use special production techniques to achieve better accuracy than typical resonators. It really isn't
Power On Reset (POR)

- Do we need a POR circuit?
  - YES – you always need some POR

- POR Options
  - Simple RC
  - Internal POR
  - External POR
Simple RC Power On Reset

- **Advantages**
  - Inexpensive
  - Simple

- **Disadvantages**
  - Very dependent on Vcc rise time
  - Not so simple

- Let’s look at an example

  Design an RC circuit for M16C/62P
A proper reset signal for an M16C/62P requires that the Reset input be held less than 0.2 Vcc for 2 mSec after Vcc reaches minimum operating voltage. The analysis of this is not easy if the power supply rises slowly so we will assume that it rises quickly compared to the 2 mSec requirement.

Using the RC formula we can calculate that one TC would allow the Reset line to rise to 0.63 times Vcc, to keep it less than 0.2 Vcc the 0.2 mSec can only represent approximately 0.2 TC. This means the circuit will have to have a 10 mSec time constant. Not too bad since we assumed an instant rise time, but in real power supplies the voltage rise is not always fast enough to be considered instantaneous.
External Power On Reset

- **Options**
  - Purchase a POR/Voltage Monitor Chip
  - Design your own

- **Purchased device**
  - Advantage
    - Simple
    - Reliable
  - Disadvantage
    - Cost
    - Must match to the MCU

If you have ruled out the RC another choice would be an external POR circuit. In this case you have two options, purchase a POR/Voltage monitor or design your own.

The advantage of purchasing a MCU monitor IC is they are pretty easy to select and connect to the MCU and they are reliable. The disadvantages are the cost and you must make sure it matches the MCU you have selected.
A second option for the external circuit is to design it. The advantage is usually considered to be the cost. The disadvantage is it can be tricky to design a very robust circuit and it usually ends up with quite a few components. The circuit shown is a fairly simple circuit which can provide an improvement over the simple RC of not being sensitive to the power supply rise time. In this circuit the zener is selected to correspond to the minimum operating voltage of the MCU. R2 sets the charge current for C1 so the slope of the charge on C1 is given by the equation shown.
The last alternative for a POW\R circuit is to select an MCU which has an internal POR. The advantages are the cost and the fact that it is already “tuned” to the MCU you have selected. The disadvantages are the MCU must have the POR feature and in most cases the POR circuit may have Vcc rise time limitations.
This is a chart from the R8C/27 circuit which shows the requirements for the internal POR circuit. Notice that there is a maximum rise time limitation on the power supply. The 20 mV per mSec time should not be a problem in most cases. If a power supply has 100 uF of capacitance it only requires 2.7 mA over the load requirement to charge the capacitance to 2.7V in the required 100 mSec. Smaller filter capacitances will reduce the current requirement. The only time this is usually a problem is when a very weak power supply or battery charge pump is used.
Low Voltage Detector (LVD) - (Brown-out Detect)

- Do I need an LVD circuit?
  - Probably

- Purpose of LVD
  - Prevent operation of MCU with Voltage < Vcc min
  - Anticipate loss of voltage
    - Save data
    - Place system in “safe” state

- LVD only monitor MCU Vcc
  - Consider all system power sources

Though you always need a POR circuit, there are cases where you may not need an LVD or “brown-out” circuit. The POR circuit prevents the device from coming out of reset it does not help when voltage dips after the device has reached the operating range. It is especially important if the device has a break in operating frequency vs. voltage.

Another use of the LVD circuit is to save data during a power loss or place the system into a “safe” state. It is important to remember that an internal LVD circuit typically only measures the MCU voltage. All the other voltages and circuits in the system must be considered to make sure that a brown-out could affect them but not the MCU.
This diagram shows just one example of using the three level voltage monitoring. In this example the Vdet2 interrupt is used to allow storing critical data to flash. When the Vdet1 level is reached the device is put into a low power state and finally if Vdet0 is ever reached the device will reset. Monitoring flags and interrupts as power is restored allows an orderly recovery of the system.
Ride Through

- Backup is not always Battery
- Example
  - Ride through 30 seconds
  - Do Not use Battery

Allow Voltage drop 3.1 to 2.8
Icc at 2 MHz = 1.5 mA

\[ I = C \frac{dV}{dT} \]
\[ C = \frac{(I \times dT)}{dV} \]

C = 0.15 Farad
* Above 0.22F @ 3.3V not common

An alternative to using a battery to ride-through a brown-out is to use a supercap. In this example we want 30 seconds of ride-through capability using a supercap. We will allow the Vcc to drop from 3.1 to 2.8V during the transient. If the R8C is running at 2 Mhz it only requires 1.5 mA but still has quite a bit of horsepower. This could even be a ride-through setting which uses the clock gearing to reduce the current during the power loss. Performing the calculation shows that we only need 0.15F to meet our design objective. The importance of low current active modes is important since values above 0.22 uF @ 3.3V are not common.
One of the features that is commonly integrated into MCU’s is the WDT. It is important to set a realistic expectation of the WDT capability. It is very useful in recovering from SW errors, however, I would not rely on it to recover from noise induced error. It will help in many cases but there is a very real possibility

An external WDT is required by some safety standards and does increase the probability of recovering from a noise incident. There are a lot of WDT/Voltage monitor circuits which provide both POR/LVD and WDT capability
Now let's look at some standard connection issues. In this circuit we are connecting a simple switch to the input of the MCU. (Poll the audience) The question is whether to use a pull-up or pull-down. Both work and there is not really a right answer but a definite consideration should be how far the switch is from the MCU. In the circuit on the left the ground connection must be sent to the switch where Vcc must be sent to the switch in the other circuit. As a general rule I would prefer to prefer to route my ground connection since it is usually the "stronger" plane and easier to find a connection to. If the switch is located off-board from the MCU I definitely would not send the Vcc to the switch without some type of protection resistor.

What about just using the internal pullups? These are OK but in many MCUs they must be enabled by software. Until they are enabled the pin is floating and there is you need to make sure the switch inputs are not used until the pullup is valid. The other issue, in some cases, is that the internal pullups are typically weak (>75K) For better noise immunity it may be desirable to have a stronger pullup or pulldown.

The last question is whether to tie the switch to an interrupt input. I only use interrupt inputs if the switch is designed to wake the MCU from a low power mode.

The reasons for not using an interrupt:

The required response time for a switch input to react to a person touching it is well over 20 mSec, based on that number the switch can be polled every 5 mSec and easily react to any touch. By scheduling the switch polling there it is easier to analyze the maximum bandwidth that will be used to service the switches and there is one less asynchronous event that has to be accounted for in other task minimum and maximum time allocations. When an interrupt is allocated to a switch I have to analyze each task and assume that it can be delayed by the switch ISR and that there could be many switch pushes in a short period of time (impatient operator syndrome)

It may seem that polling the switches though they will not usually be pushed is a waste of time. However, polling a switch typically takes less than 2 uS to see if it is active. If that task is done every 5 mSec the total MCU BW used to poll a switch is 0.04% of the MCUs available BW (roughly the equivalent of consuming 1 second out of an hours worth of work)

The last item to consider about monitoring switches is that a switch requires debouncing. In a polled condition it is very easy to check the status 4 consecutive times and if they are all represent a change of state then implement the action. With an interrupt special routines must be used to ignore subsequent “bounce” interrupts and to reject noise. This additional code and overhead is often quite a bit more complex than the simple polling. The WORST thing is the 20 mSec delay loop in the ISR to ensure a second switch event is not registered due to bounce.
A question that often comes up is how large can a pullup be, this is especially important in low power applications when the input may be active for quite a bit of time or if the pullup is used on a line that may be later configured as an output and driven to the opposite state of the pull resistor. The slide shows a typical calculation.
This slide shows one solution that can be used to overcome the power loss of the pull up resistor. In this case a port pin is used though an external FET could be used for a lower rdsON. In this case the output is configured high only long enough to monitor the state of S1, it is then driven low to minimize the power loss. As shown before the time required to monitor the input is only 0.04% duty so the total energy expended is extremely low.
Level Translation

- Problem - Interfacing a 3V micro design to 5V LCD
  - Writing to LCD

ACM0802B SERIES LCD MODULE

4.0 ELECTRICAL CHARACTERISTICS (Ta = 25°C, VDD = 5.0V ± 0.25V)

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Condition</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply voltage for LCD drive</td>
<td>VDD-V0</td>
<td>Ta = 25°C</td>
<td>4.3</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Input voltage</td>
<td>VI</td>
<td>-</td>
<td>4.75</td>
<td>5.0</td>
<td>5.25</td>
<td>V</td>
</tr>
<tr>
<td>Input high voltage</td>
<td>VIH</td>
<td>-</td>
<td>2.2</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Input low voltage</td>
<td>VIL</td>
<td>-</td>
<td>0.0</td>
<td></td>
<td>0.6</td>
<td>V</td>
</tr>
<tr>
<td>Output high voltage</td>
<td>VOH</td>
<td>IOH=0.2mA</td>
<td>2.4</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Output low voltage</td>
<td>VOL</td>
<td>IOL=1.2mA</td>
<td>-</td>
<td></td>
<td>0.4</td>
<td>V</td>
</tr>
<tr>
<td>Supply current</td>
<td>ICC</td>
<td>VDD=5.0V</td>
<td>-</td>
<td></td>
<td>3.0</td>
<td>mA</td>
</tr>
<tr>
<td>Input leakage current</td>
<td>IIL</td>
<td>-</td>
<td>-</td>
<td></td>
<td>1.0</td>
<td>μA</td>
</tr>
<tr>
<td>LED power supply current</td>
<td>LED</td>
<td>VLED=5V, R0=4.7kΩ</td>
<td>-</td>
<td>170</td>
<td></td>
<td>mA</td>
</tr>
</tbody>
</table>

- R8C Voh is (Vcc – 0.5V) @ 5 mA
  - No Problem

A common problem in connecting IO devices is that they must run at a different Vcc level than the MCU. In this example I want to interface an LCD module to the MCU. Though the R8C can run at different Vcc levels I may be limited by some other consideration, for example, the voltage rating limitation on the supercap circuit we previously looked at. Looking at the LCD specification we can see that writing to the LCD is not actually a problem. The minimum input voltage for the LCD is only 2.2V and the R8C can easily provide that level even when it is running at 3V.
What might be an apparent problem is that the R8C inputs are not 5V tolerant. However, this is really not a huge issue. By placing a voltage divider as shown in the figure we divide down the 5V signals from the LCD to the MCU but do not affect the signals from the MCU to the LCD (assuming it has a high input impedance).
Unfortunately, going the other way is not quite as easy, though not difficult by any means. In this case the MCU requires 0.8 * Vcc or 4V minimum for a valid high. In this case a transistor circuit can be used. The logic will be inverted but that is easily handled in software. In this case R2 is chosen to limit the sensor current within its rating. R1 is chosen to limit the power consumption in the on-state and minimize noise effects. A minimum value is given by current through R2 time the minimum Beta of the transistor divided by 5V (5V/(I_R2*Beta))
A Power Output

- Designing a simple power drive
  - Get Load requirement
  - Divide by port output current
  - This gives minimum hFE or Beta
- \( R_2 = \frac{V_{out} - 0.6}{\text{rated output current of port pin}} \)
- D1 rated at load current

The diagram and notes show how easy it is to do a power interface from an MCU. For example, if you need to drive a 12V relay that requires 300 mA the following procedure picks the components. 300 mA/5mA gives a transistor beta requirement of 60. This can easily be accomplished by most signal transistors equivalent to the 2N2222. If the beta requirement gets higher then a darlington can be used, these devices typically have Beta’s greater than 1000.
A common decision is what to do with unused inputs. The HW manual should always be checked for specific requirements but in many cases the HW manual will just indicate that you should pull it one way or the other or set it as an output. My preference is to pull low. Ground connection are usually easier to find and more solid. The other option is to set the pin as an output. The imitations with this are that the pin is floating as an input until it is set to an output. This means it has higher vulnerability to noise and can draw some extra current until set. This extra current is typically not more than 1 mA and does not cause an issue but it does add up. I prefer to set the output low only since it then connects to the ground plane.
Questions?
Summary

- Selecting clock circuit
- POR/LVD
- WDT requirements
- Input Circuits
- Output Circuits
One of the most significant innovations that has changed the embedded world is the tremendous amount of peripheral integration into a single MCU device. Analog and digital circuits that used to be external devices are not being pulled into one chip, eliminating the need for complex external bus systems, translators and glue logic. It's almost like you can just drop the chip down and start writing software.
To Interrupt or Not

- Probably Not
  - Switches – except for low power wake-up
  - A/D
  - SPI

- Probably
  - UART Receive/Transmit
  - Timers
  - Pulse counting or edge detection
Polled Switch Routine

- Use Timer Tick for scheduling
  - Setup below for 1 mSec tick
  - Samples switch every 5 mSec

```c
if ((tick_timer - last_sample_time)>4)){
    if (SW1 )
        SW1_count++;
    else
        SW1_count = 0;
    if (SW1_count > 5){
        SW1_state = ACTIVE;
    }
    last_switch_sample = tick_timer;
```
External Power On Reset (Cont)

- Assume Q beta = 75
- $V_{cc} = 5V$, $V_{min} = 2.7V$
- Set $C_1$ to 0.22 uF
- Charge time 10 mSec for 2.7V charge
- $I = C \frac{dv}{dt} = 0.22 \text{ uF} \times 0.5V/2\text{mS} = 50 \text{ uA}$
- $R_1$ sets zener current. Typical 0.5 mA current would need 10K
- $R_2 = (2.7V - 0.6)/50 \text{ uA} = 42K$
- $R_3$ just a discharge path 100K

Zener set for MCU $V_{min}$

$R_3 \gg R_2$

Reset Line slope $dV/dT$ is approximately

$\left[\frac{(V_z-0.6)}{R_2}\right]/C_1$

This slide shows a design calculation. The value for the charge current for $C_1$ uses 0.5V as a limit since this would be less than 0.2*$V_{min}$ so it should be a conservative calculation.