A low-cost SoC power measurement platform

Evaluating software impact on power consumtion is not easy. Using a simple multimeter on the power line is not accurate enough because there are big variation in input current. A higher sampling frequency is needed. Also you have to correlate the power data with the various phases of a software (separate initialisation from the computation part for example).

Our idea is to build a low-cost power measurement tool around a cheap microcontroller, in this case from the pic18 family. We use its ADC to get current and voltage sample of the power line at a high enough rate. We also use its GPIOs as input, connected to GPIOs of the SoC being evaluated, to track and report various software state on the SoC.

We choose to use a pic18fj53. Its has a 12 bits ADC which can do up to 80ksp/s, runs at 48Mhz and has a USB interface. Our plan was to use the USB interface to transfer the data to the host.Unfortunably during developement we found that the USB interface, when enabled, would dramatically decrease the ADC's accuracy. So we choose to use the serial port on the PIC side and added an external USB/serial bridge. The electronic gets power from the USB connection, the external power supply being used only for the SoC being monitored.

The requirement are:

- inputs between 0 and 2A, 0 and 20V
- 4 GPIOs to connect with the SoC being evaluated

A 1.350V precision diode is used as a voltage reference for the ADC. The input voltage is feed to ADC input 0 (AN0) via a voltage diviser. To cover the input range with enough accuracy 3 different values can be selected with a jumper.

3 Shut resistors are used for current measurements, selectable with a jumper. An amplifier will multiply the shut's voltage before feeding it to AN1.

RB0-RB3 are used to get 4 GPIO status from the SoC.

Each sample reported to the host is composed of 3 values: GPIO state (integer between 0 and 15), voltage (AN0 value between 0 and 4095) and current (AN1 value between 0 and 95), in the format: *gvvvcccX*, with *g*, *v* and *c* being hexadecimal digits and *X* being the letter X used as separator. With 8 bytes per sample, at 921600 bauds we could report in theory up to 11520 samples/s. To be safe we choose to run at 5000 samples/s (which requires 10000 A/D conversions/s).