Introduction to the LPC1114.

Core architecture

Key terms: 32 bit, Harvard, registers, arithmetic units, instruction set, instruction decoder, fetch, decode, execute, machine code versus assembly language and higher level languages

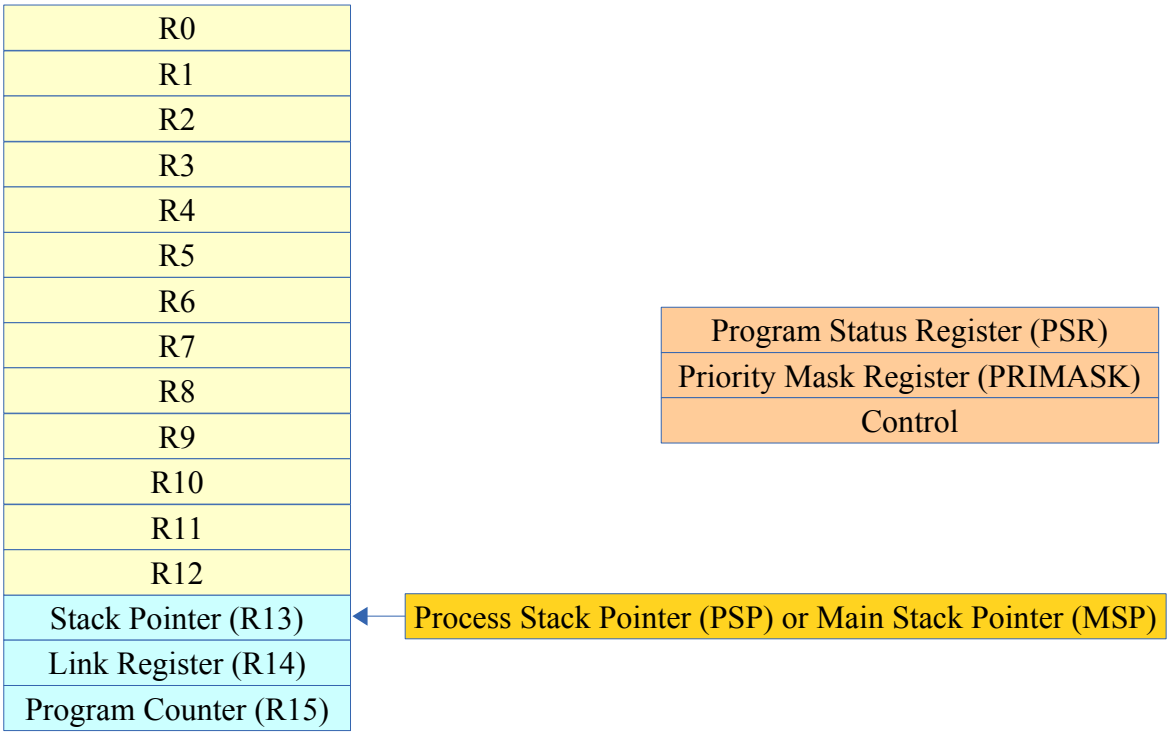
The LPC1114 is a microcontroller incorporating an ARM Cortex M0 microprocessor core, some memory and some input/output circuits.

ARM Ltd is a microprocessor design company based in Cambridge in England. It licenses designs to many manufacturers around the world. There are many billions of ARM microprocessors in the world today.

The Cortex M0 microprocessor core can be found in many microcontrollers from different companies (e.g NXP, Atmel, ST-Microelectronics, Texas Instruments). It is a 32 bit processor which means that its internal calculations are all performed with 32 bit resolution (by default). The LPC1114 we use can operate at 50MHz which allows it execute up to 50 million instructions per seconds (MIPS). In a world with multi-core, multi-gigahertz desktop and mobile microprocessors this might not seem very fast however in the embedded world this is sufficient for controlling and monitoring most real-world equipment.

The Cortex M0 Core

M0 Registers



All registers are 32 bits wide. R0 to R12 are general purpose registers. R13 is the Stack Pointer which is used to manage a RAM based descending stack. The stack is used for local variables within functions, backup copies of registers, parameter passing and for storing function call return addresses. The stack pointer is actually one of two registers. Unprivileged programs use the Process Stack Pointer which Operating systems (and support functions) make use of the Main Stack Pointer. The switch between stack pointers can be handled automatically.

R14, the Link Register contains the address a program must return to when it completes the current function.

R15 is the Program Counter which points to the next machine code instruction to be fetched and executed.

In addition to R0-R15 there are three more core registers. These are accessed using special machine instructions and they control the way in which the processor operates.

The program status register (PSR) contains arithmetic flags which are central to every microprocessor's ability to make decisions. Additional fields are useful for interrupt handlers and operating systems.

The Priority Mask register (PRIMASK) is used by the Cortex M0 to enable/disable maskable interrupts

The Control register is used to select which of the stack pointers is currently active and also to set the current privilege level.

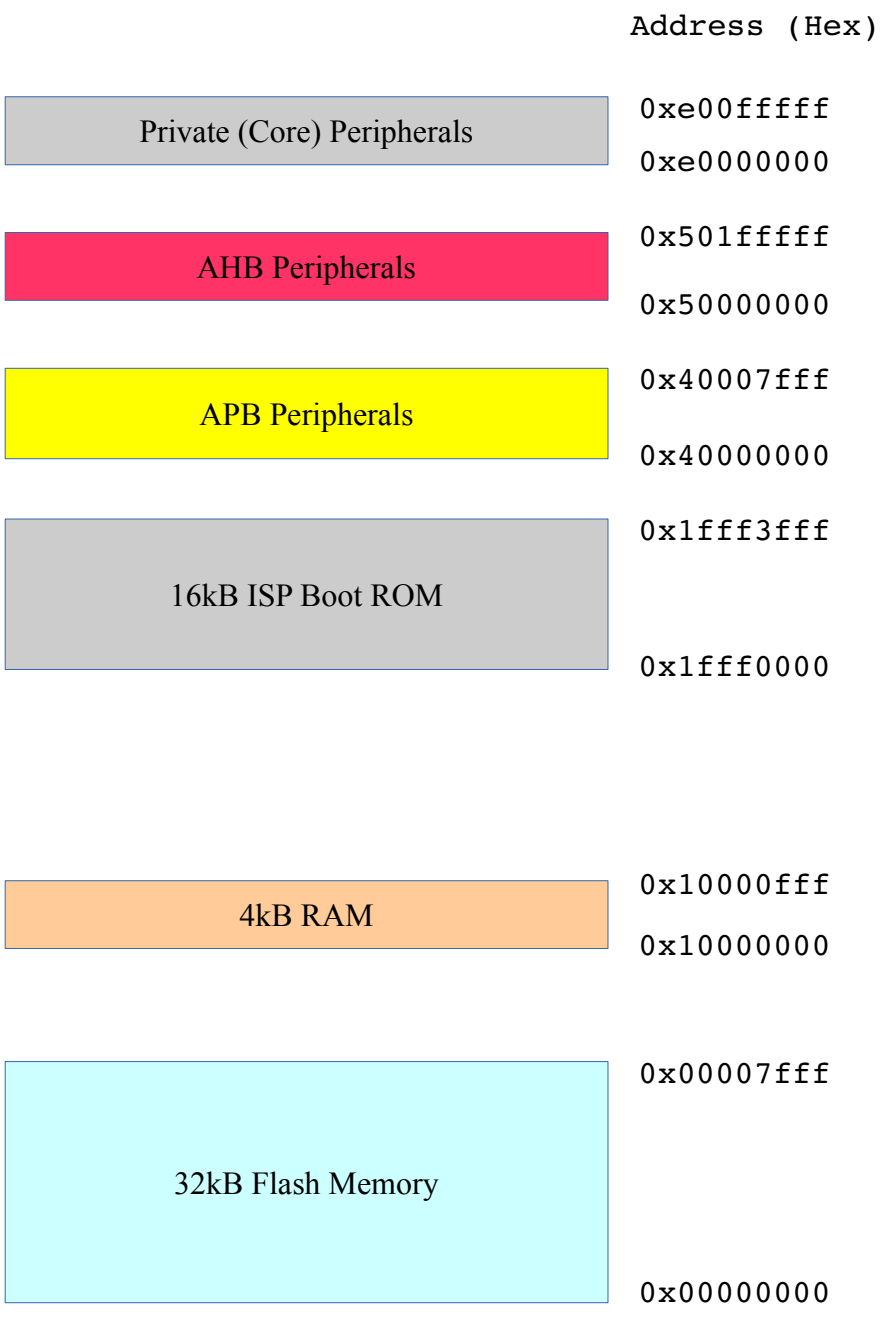
Instruction set summary

The Cortex M0 executes the Thumb 2 Instruction set. A summary of the instruction set is available on the course website (see Thum2ReferenceCard.pdf)

LPC1114 Memory map

Key terms: Flash, ROM, memory mapped IO, 4GB, ISP Bootloader, Buses, APB, AHB

There are a few different version of the LPC1114. The one we use is the LPC1114FN28/102. This has 4kB of RAM and 32kB of Flash (program) memory. The device also incorporates a number of sub systems referred to as peripherals (even though they are inside the chip). These peripherals provide an interface to the outside world and make this device a ***microcontroller*** rather than a ***microprocessor***. The memory map (what device is at which address) is shown below. These devices are connected to the CPU core over 1 of 2 sets of conductors (buses). Higher speed devices such as memory and parallel digital I/O are connected over the AHB (Arm High-performance Bus). Lower bandwidth devices are connected over the APB (Arm Peripheral Bus) such as the Analogue to Digital Converter.



Programs are stored in the 32kB Flash memory area. The contents of this memory are **not lost** when powered off. Data and stack are located in the 4kB of RAM. The contents of RAM are lost when powered off.

The LPC1114 can be programmed after it has been deployed in the field. This is called In-System-Programming (ISP) and it is the mechanism that we will use to program the device (there is another, more sophisticated way but it is a little more complex). The code that manages ISP is activated when one of the device pins is shorted to 0V at boot up. This code is referred to as the BOOT ROM.

The figure below, taken from the NXP LPC1114 Reference Manual (Document UM10398) illustrates the relationship between the core, the peripherals and buses in the LPC1114