The high performance of the PIC18CXXX devices can be attributed to a number of architectural features commonly found in RISC microprocessors. These include:

- Harvard architecture
- Long Word Instructions
- Single Word Instructions
- Single Cycle Instructions
- Instruction Pipelining
- Reduced Instruction Set
- Register File Architecture
- Orthogonal (Symmetric) Instructions

**Harvard Architecture:**
Harvard architecture has the program memory and data memory as separate memories, which are accessed from separate buses. This improves bandwidth over traditional von Neumann architecture in which program and data are fetched from the same memory using the same bus. To execute an instruction, a von Neumann machine must make one or more (generally more) accesses across the 8-bit bus to fetch the instruction. Then data may need to be fetched, operated on and possibly written. As can be seen from this description, the bus can become extremely congested. With a Harvard architecture, the instruction is fetched in a single instruction cycle (all 16 bits). While the program memory is being accessed, the data memory is on an independent bus and can be read and written. These separated busses allow one instruction to execute, while the next instruction is fetched. A comparison of Harvard and von Neumann architectures is shown below.

**Long Word Instructions:**
Long word instructions have a wider (more bits) instruction bus than the 8-bit data memory bus. This is possible because the two buses are separate. This allows instructions to be sized differently than the 8-bit wide data word and allows a more efficient use of the program memory, since the program memory width is optimized to the architectural requirements.

**Single Word Instructions:**
Single word instruction opcodes are 16-bits wide making it possible to have all but a few instructions be single word instructions. A 16-bit wide program memory access bus fetches a 16-bit instruction in a single cycle. With single word instructions, the number of words of program memory locations equals the number of instructions for the device. This means that all locations are valid instructions.

Typically in the von Neumann architecture, most instructions are multi-byte. In general, a device with 4 Kbytes of program memory would allow approximately 2K of instructions. This 2:1 ratio is generalized and
dependent on the application code. Since each instruction may take multiple bytes, there is no assurance that each location is a valid instruction.

**Double Word Instructions:**
Some operations require more information then can be stored in the 16 bits of a program memory location. These operations require a double word instruction, and are therefore 32-bits wide.

Instructions that require this second instruction word are:

- Memory to memory move instruction (12 bits for each RAM address)
  - MOVFF SourceReg, DestReg
- Literal value to FSR move instruction (12 bits for data and 2 bits for FSR to load)
  - LFSR FSR#, Address
- Call and goto operations (20 bits for address)
  - CALL Address
  - GOTO Address

The first word indicates to the CPU that the next program memory location is the additional information for this instruction and not an instruction. If the CPU tries to execute the second word of an instruction (due to a software modified PC pointing to that location as an instruction), the fetched data is executed as a NOP.

Double word instruction execution is not split between the two TCY cycles by an interrupt request. That is, when an interrupt request occurs during the execution of a double word instruction, the execution of the instruction is completed before the processor vectors to the interrupt address. The interrupt latency is preserved.

**Instruction Pipeline:**
The instruction pipeline is a two-stage pipeline that overlaps the fetch and execution of instructions. The fetch of the instruction takes one TCY, while the execution takes another TCY. However, due to the overlap of the fetch of current instruction and execution of previous instruction, an instruction is fetched and another instruction is executed every TCY.

**Single Cycle Instructions:**
With the program memory bus being 16-bits wide, the entire instruction is fetched in a single machine cycle (TCY), except for double word instructions. The instruction contains all the information required and is executed in a single cycle. There may be a one cycle delay in execution if the result of the instruction modified the contents of the program counter. This requires the pipeline to be flushed and a new instruction to be fetched.

**Two Cycle Instructions:**
Double word instructions require two cycles to execute, since all the required information is in the 32 bits.

**Reduced Instruction Set:**
When an instruction set is well designed and highly orthogonal (symmetric), fewer instructions are required to perform all needed tasks. With fewer instructions, the whole set can be more rapidly learned.

**Register File Architecture:**
The register files/data memory can be directly or indirectly addressed. All special function registers, including the program counter, are mapped in the data memory.

**Orthogonal (Symmetric) Instructions:**
Orthogonal instructions make it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of “special instructions” make programming simple yet efficient. In addition, the learning curve is reduced significantly. The Enhanced MCU instruction set uses only three non-register oriented instructions, which are used for two of the cores features. One is the SLEEP instruction, which places the device into the lowest power use mode. The second is the CLRWDT instruction, which verifies the chip is operating properly by preventing the on-chip Watchdog Timer (WDT) from overflowing and resetting the device. The third is the RESET instruction, which resets the device.

The information in this document is obtained from PICmicro 18C MCU Family Reference Manual. (20050725)