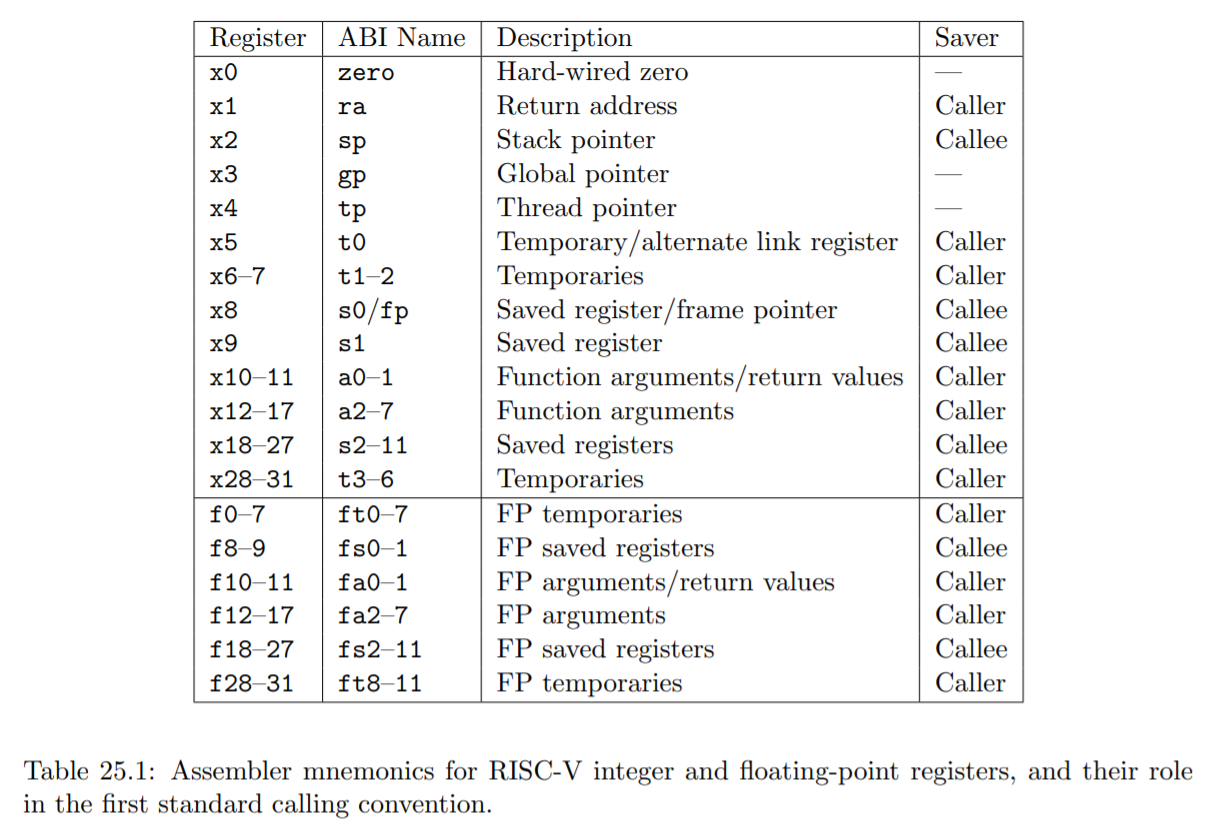
**從 RISC-V 處理器到 UNIX 作業系統閱讀筆記暨心得**

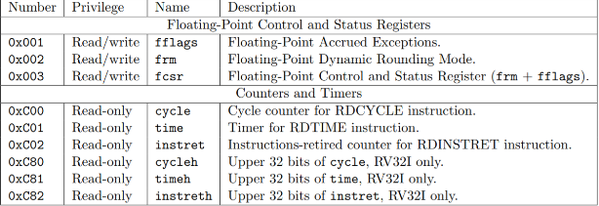
**RISC-V處理器**

RISC 架構有以下特點:

* 指令長度固定，CPU 更容易做解碼
* 盡可能將資料放在 CPU 的暫存器中操作
* 處理器的硬體簡化，所以時脈容易提高且更省電
* 因為複雜的指令皆由多個基本指令組成，對於記憶體的開銷更大

RISC-V 處理器中的通用暫存器:

**控制暫存器**

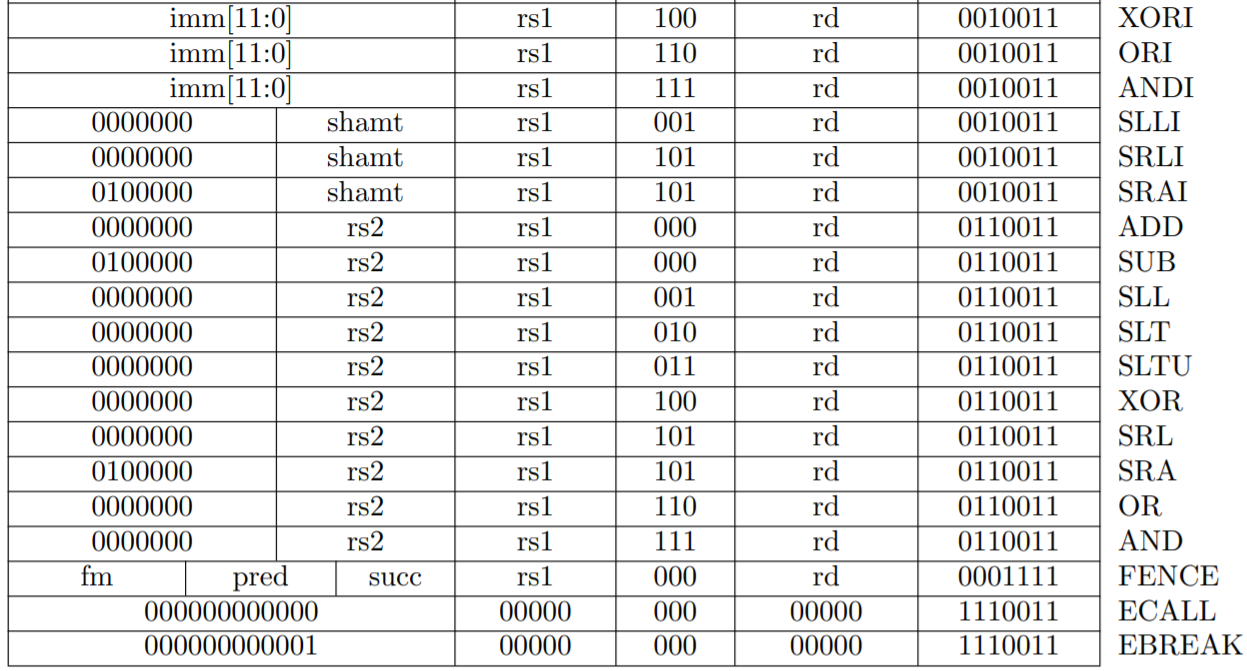


CSR 暫存器在設計作業系統時會很有用，可以透過讀取 time 得到目前的時間值。

**指令集**

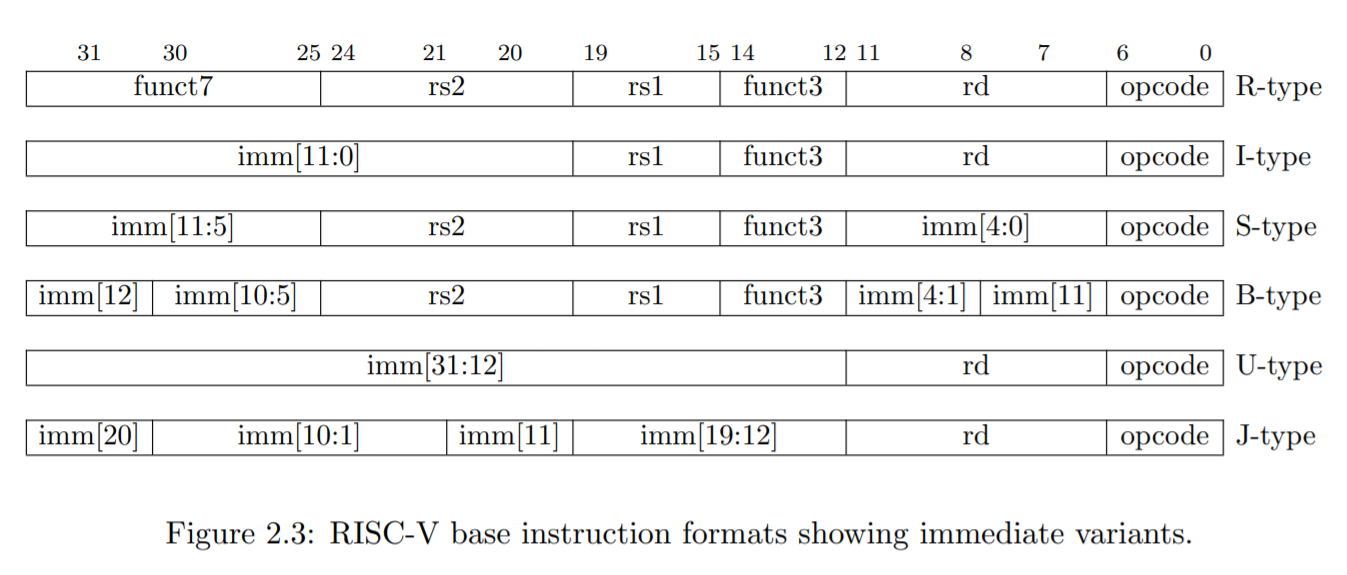
* RV32I：基础整数指令集 (固定不變了)
* RV32M：乘法和除法
* RV32F：單精度浮点操作（和 RV32D:雙精度浮點操作）
* RV32A：原子操作
* RV32C：可选的压缩扩展 (對應 32 位元的 RV32G)
* RV32B：基本擴展。
* RV32V：向量扩展（SIMD）指令
* RV64G：RISC-V 的 64 位地址版本。



RV32I 是必備的指令集，其指令與格式如下所式:

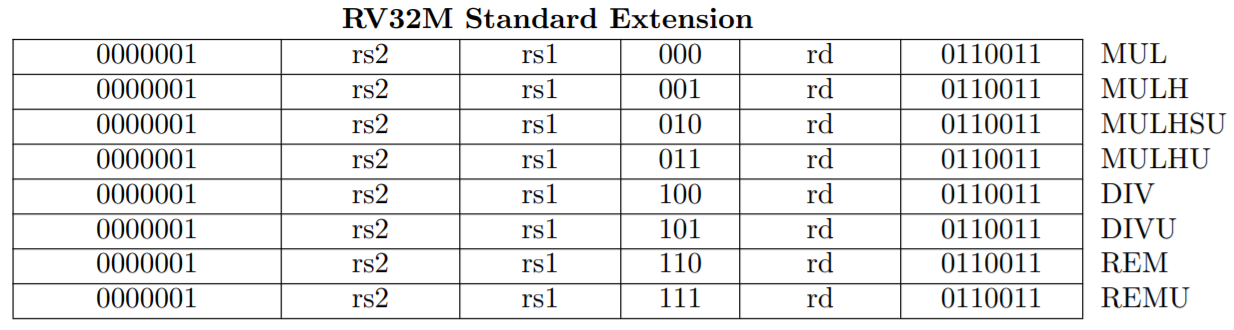
## **RISC-V 指令格式**

RISC-V 的指令格式共有六種



imm 被分為更多部分,主要為了相容與擴充性,Verilog 或 VHDL 設計電路時可以用 decoder 先將立即值組成單一欄位,然後開始執行指令。

**RV32M 的乘除法指令格式如下**

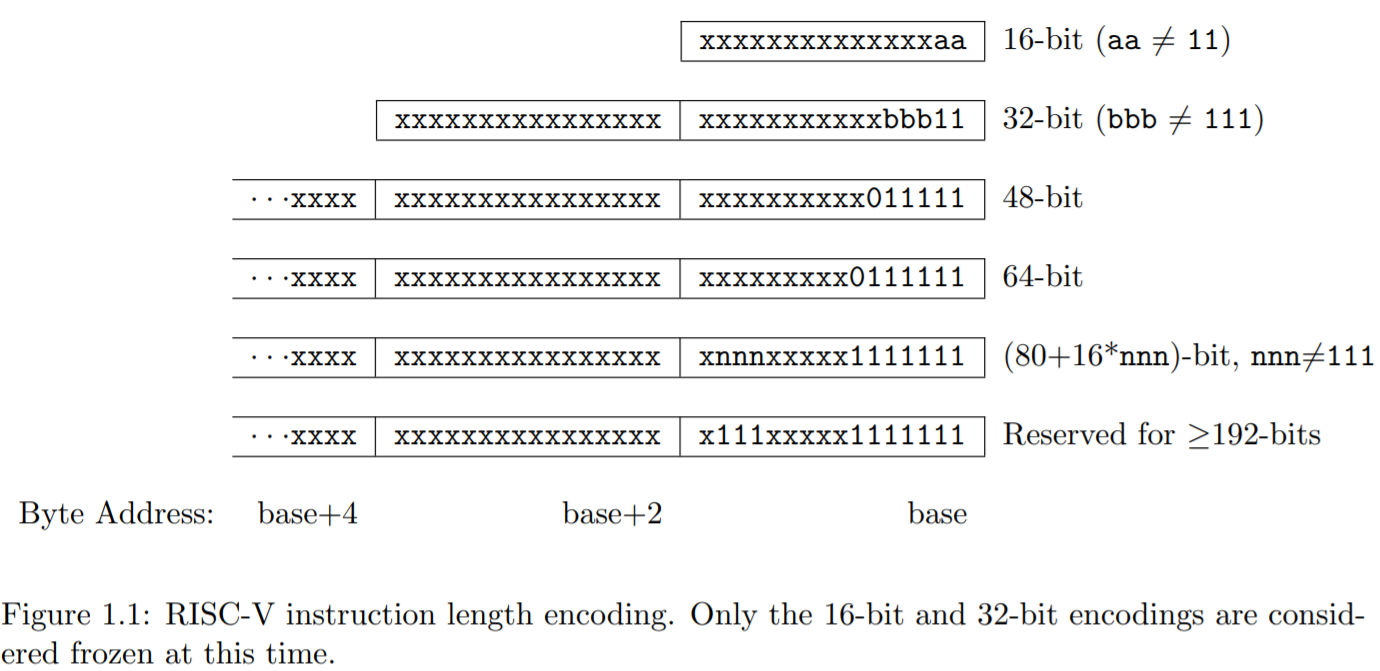


RV32M 由兩個 32 位元整數相乘可能變成 64 位元整數,因此可能超過 32 位元的高半部,所以有號數的乘法也分為 MUL, MULH 這兩種

無號數的乘除法指令,會以 U 結尾，比如MULHSU, MULHU, DIVU, REMU 等都是無號數的乘除法指令,除了 REM 代表取餘數 (Remainder)

## **壓縮指令集**

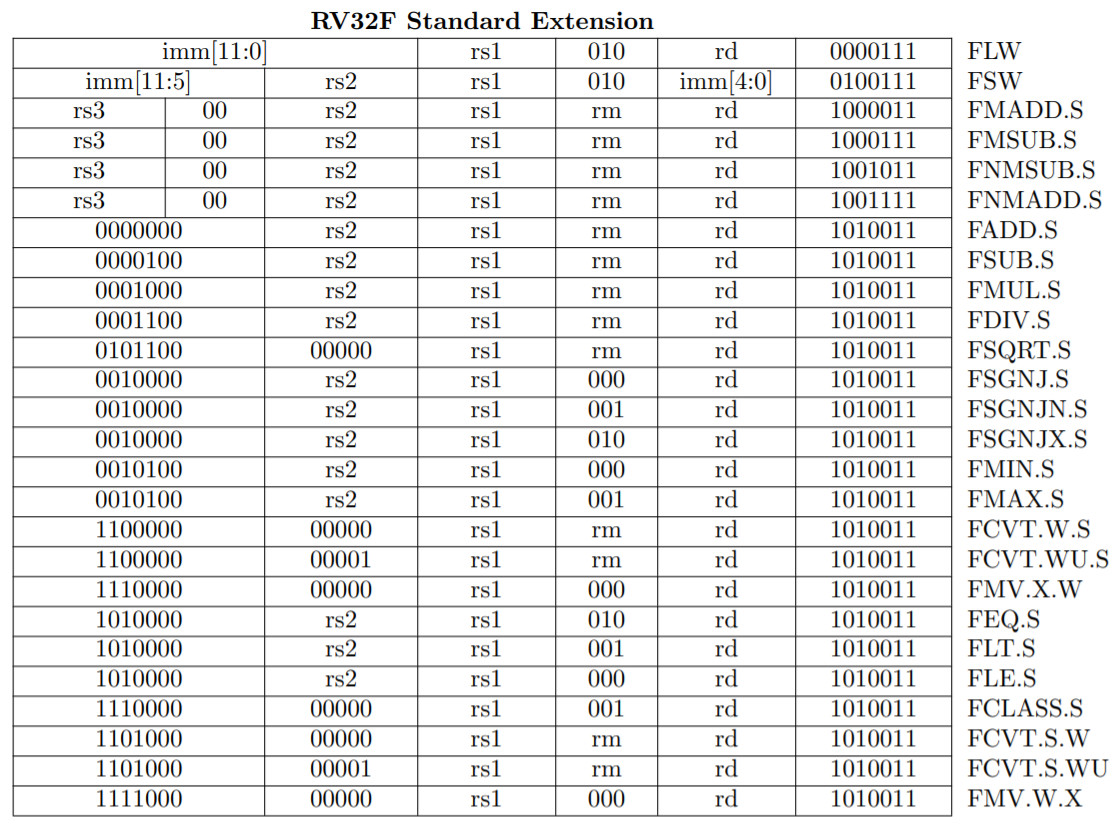
指令集支援了壓縮模式 RV32C，很多指令可以被壓縮表達成 16 位元模式，只要最低的兩個位元 aa 不是 11，你便是壓縮的 16 位元指令。



如果 aa=11但bbb ！= 111，那便是 32bit 指令,RISC-V 會有 16, 32, 48, 64 等格式的指令,保留更長指令的編碼空間

## **浮點指令集**

單精度浮點指令 RV32F 的格式表



雙精度 RV32D 的格式