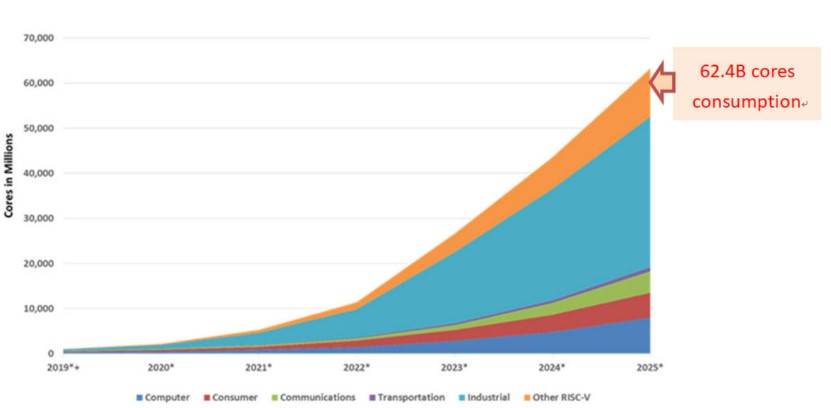
˙RISC-V

RISC-V 結合了過去數十年的指令集基礎開發出的一種精簡指令集中的開源硬體架構，他的目標是追求簡潔性、低成本以及高性能，希望成為一個通用的指令集架構，比起其他匯編語言，他的ISA基礎模塊化的特徵使得其開放性極高，方便進行編程、編譯與鏈接，不受到未來趨勢發展的影響，達成兼容、穩定的主要性能架構，根據台灣務聯網產業技術協會的資料統計與預測(如圖一)，這將帶來更大的優勢， RISC-V的指令數量的使用與成長速度自2019年至2025年快速增長，在現今極為活躍，挑戰主流專有ISA的主導地位，由以下進行分析與介紹。



(圖一。源自http://www.twiota.org/eventDetails.aspx?id=3cda72c0-bd73-4085-ad71-c6b8063afdc2)

˙RV321

RV321作為RISC-V的基本整數指令集，相較於以往的指令集，能夠提升計算性能，擁有其獨特的優點，由以下六點介紹:

1. 指令格式:RV321總共有47道指令，一共可分為六大類指令，分別是:
   1. I-Type­—使用於短立即數和訪存load操作
   2. R-Type—使用於寄存器與寄存器之間操作
   3. S-Type—使用於訪存store操作
   4. B-Type—使用於條件跳轉操作
   5. U-Type—使用於長立即數跳轉操作
   6. J-Type—使用於無條件跳轉操作

RV321僅使用了32位指令字中不到八分之一的編碼空間，為ISA擴展出足夠的空間，簡化了數據通路的控制邏輯，同時大幅降低了成本。

1. 寄存器:RV321僅須由零寄存器作為操作數，便能夠完成其他缺少零寄存器的指令集可以完成的操作，相較於分支預測複雜的PC，其分支指令預測的準確性較不被擔心。
2. 整數計算: RV321提供了單獨的移位指令，不包含乘法與除法，以低寬數的數據訪問減少大量能量，RISC-V不需要額外添加成除法擴展便能夠完整運行，同時縮小了嵌入式晶片的面積。
3. Load和Store: RV321能夠容納任何數據類型，在移植舊代碼時邊界對齊的訪問要求較為寬容，簡化了整體設計，達成高開放性的目標。
4. 條件分支: RV321透過去掉部分擁有延遲分支特性的指令集，避免了條件碼需要設置隱式狀態的額外負擔，達成了簡潔化的目標。
5. 無條件跳轉: RV321可以利用將零寄存器替換ra作為目標寄存器，達成無條件跳轉的目標，避開了錯綜複雜的程序調用指令，以達成簡潔化的目標。

以上RV321的內容與特徵與ARM-32、MIPS-32、x86-32等指令集相比，主要襯托與強調了RISC-V簡約與排除過去多項瑣碎錯誤的設計，促進未來指示令在編寫與運用的發展。

˙RISC-V彙編語言