

# instruction format

الـ instruction format يحدد تخطيط بتات التعليميات (bits of an instruction) من حيث الحقول المكونة لهاذي التعليمات. الـ instruction format يجب ان يتضمن opcode و implicitly و explicitly و صفر operand او اكثر من operands .

تتم الاشارة الى كل explicit operand باستخدام احد الـ addressing modes . بالنسبة لمعظم الـ instruction sets يتم استخدام اكثر من instruction format . الـ instruction format يعد تصميم . complex art

## Instruction Length

المشكلة الاساسية في التصميم هي طول التعليمات. القرار هاض ياثربحجم الذاكرة (memory size) وتنظيم الذاكرة (memory organization) وبنية الناقل (bus structure) و تعقيد المعالج (processor complexity) وسرعة المعالج (processor speed).

الـ trade-off يتكون بين رغبة الحصول على تعليمات قوية و توفير مساحة. المبرمجون يريدون المزيد من الـ opcodes والمزيد من الـ operands والمزيد ايضا من addressing modes ونطاق عناوين اكبر. يعني opcodes اكثر و operands اكثر بتخلي تعامل المبرمج اسهل لان يمكن كتابة برنامج اقصر لينجز مهمة معينة. وايضا من ناحية الـ addressing modes تمنح المبرمج مرونة اكبر في تنفيذ وظائف معينة مثل branching. وطبعاً مع زيادة حجم main memory والاستخدام الزائد لـ Virtual Memory المبرمجين يريدون ان يكونو قادرين على معالجة نطاقات اكبر من الذاكرة.

كل هاي المواضيع ( opcodes, operands, addressing modes, address range ) تتطلب وحدات بت وبتخلي instruction lengths اطوال . ممكن ان يكون طول التعليمات الاطوال مضيعة للوقت. الـ 64-bit instruction bit تشغل ضعف الـ 32-bit instructions ولكنها على الأرجح اقل فائدة منها في بعض الحالات ممكن ان يتم التعامل مع البيانات تستخدم 16-bit . بعيدا عن موضوع الـ trade-off اما يكون الطول مساويا لـ لطول الـ bus system و databus length او يكون احد منهم ضعف الاخر. والا فلن نحصل على integral number من التعليمات في fetch cycle .

اعتبار له صلة هو الـ memory transfer rate ان لم يواكب هذا المعدل الزيادة في الـ processor speed .

من الممكن ان يحدث memory bottleneck .

## Allocation of Bits

بقى مشكلة وهي كيفية تخصيص البتات ؟. الـ trade-offs هون تعتبر اعقد. Opcodes اكثر يعني بتات في الـ opcode field اكثر. بالنسبة لـ instruction format بطول محدد يقلل عدد البتات المتاحة للعبارة. هناك تحسين واحد للـ trade-off هو استخدام opcodes بطول متغير ( variable-length opcodes ). في هذا الاسلوب يوجد حد ادنى لطول الـ opcode ولكن بنسبة لبعض opcodes قد يتم تحديد عمليات اضافية باستخدام بتات اضافية في التعليمات. بالنسبة لتعليمات ذات الطول الثابت (fixedlength instruction) فان هذا يتكرر عدد اقل من البتات القابلة للعبارة. يتم استخدام هذه الميزة لتعليمات التي تتطلب عدد اقل من operands و/او اقل powerful addressing .

العوامل هاذي تدخل في تحديد استخدام addressing bits :

- Number of addressing modes

في بعض الاحيان يمكن النظر في addressing mode . على سبيل المثال بعض الـ opcodes دائما تتطلب الـ indexing وفي اوضاع اخرى يجب ان تكون addressing modes تكون explicit او الحاجة الى بته واحدة او اكثر من الـ mode bits .

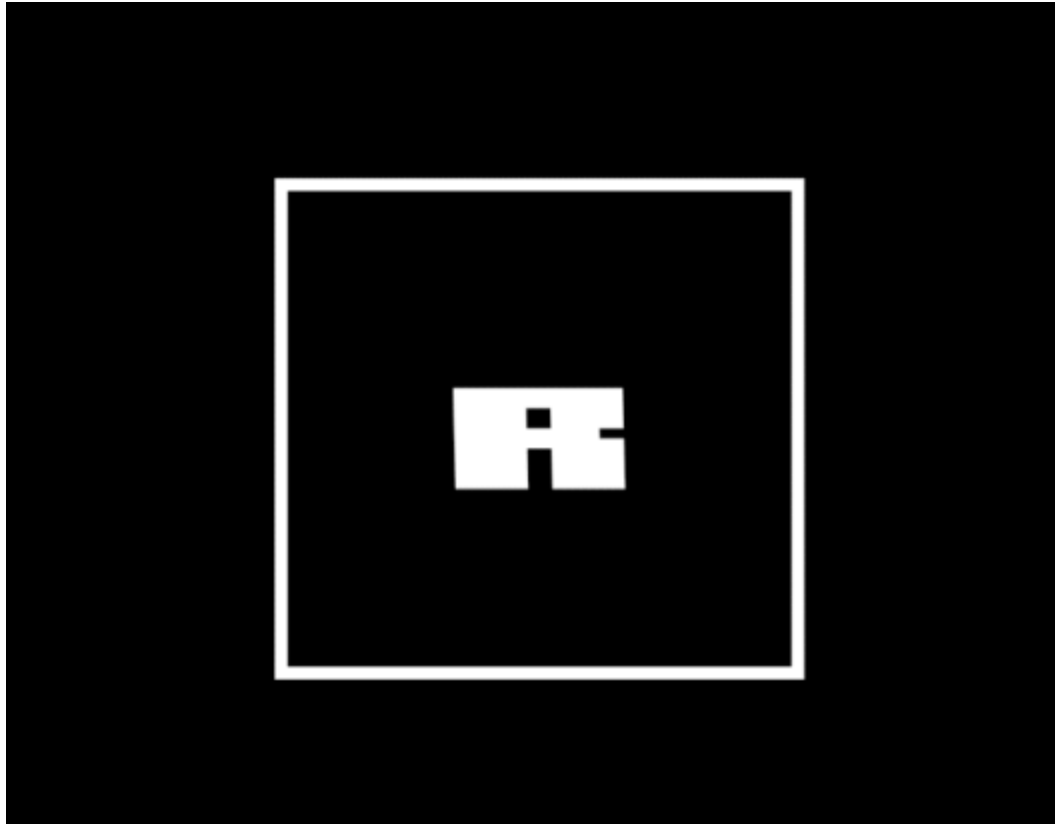
- Number of operands

عدد اقل من العناوين يؤدي الى برنامج اطول واكثر تعقيدا. تشتمل الـ instruction formats في الاجهزة الحالية على two operands او اكثر. قد يتطلب كل operand address في التعليمات مؤشر للوضع الخاص به.

الثانوي :

- Register versus memory
- Number of register sets
- Address range
- Address granularity

**AhmadAlFareed**



Twitter : [https://twitter.com/dr\\_retkit](https://twitter.com/dr_retkit)

YouTube : <https://www.youtube.com/@retkit1823>