

Kingdom of Saudi Arabia
Ministry of Education
Imam Abdulrahman bin Faisal University
College of Science and Humanities in Jubail
Computer Science Department

المملكة العربية السعودية
وزارة التعليم
جامعة الإمام عبدالرحمن بن فيصل
كلية العلوم والدراسات الإنسانية بالجبيل
قسم علوم الحاسب



جامعة الإمام عبد الرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

ثمانية أفكار عظيمة في هندسة معمارية الحاسب

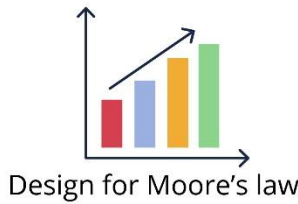
مقرر معمارية الحاسب الآلي

أريج أحمد صالح

لولوة وليد الضويحي

في هذا المقال نقدم ثمانية أفكار رائعة اخترع فيها مهندسو الكمبيوتر آخر 60 عامًا من تصميم الكمبيوتر. هذه الأفكار قوية لدرجة أنها استمرت بعد فترة طويلة من أول جهاز كمبيوتر يستخدمها، مع توضيح المهندسين المعماريين الجدد إعجابهم بتقليد أسلافهم.

1- تصميم لقانون مور Design for Moore's law



العامل الثابت الوحيد لمصممي الكمبيوتر هو التغيير السريع، والذي يحركه بشكل كبير قانون مور. تنص على أن موارد الدوائر المتكاملة تتضاعف كل 18-24 شهرًا. نتج قانون مور عن تنبؤ عام 1965 يمثل هذا النمو في سعة الدوائر المتكاملة بواسطة جوردون مور، أحد مؤسسي شركة إنتل. نظرًا لأن تصميمات الكمبيوتر قد تستغرق سنوات، فإن الموارد المتاحة لكل شريحة يمكن أن تتضاعف بسهولة أو تتضاعف أربع مرات بين بداية المشروع ونهايته. مثل مطلق النار على الجليد، يجب أن يتوقع مهندسو الكمبيوتر أين ستكون التكنولوجيا عندما ينتهي التصميم بدلاً من التصميم من حيث تبدأ. نستخدم الرسم البياني لقانون مور "الأعلى وإلى اليمين" لتمثيل التصميم من أجل التغيير السريع.

2- استخدام التجريد لتبسيط التصميم Use Abstraction to Simplify Design

كان على كل من مهندسي الكمبيوتر والمبرمجين أن يبتكروا تقنيات لجعل أنفسهم أكثر إنتاجية، وإلا فإن وقت التصميم سوف يطول بشكل كبير مع نمو الموارد بموجب قانون مور. تتمثل إحدى تقنيات الإنتاجية الرئيسية للأجهزة والبرامج في استخدام الأفكار المجردة لتمثيل التصميم على مستويات مختلفة من التمثيل؛ يتم إخفاء تفاصيل المستوى الأدنى لتقديم نموذج أبسط في المستويات الأعلى. سنستخدم رمز الرسم التجريدي لتمثيل هذه الفكرة الرائعة الثانية.

3- جعل الحالة العامة سريعة Make the Common Case Fast

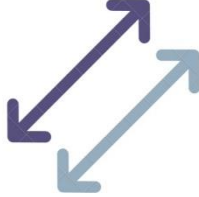
سيؤدي جعل الحالة الشائعة سريعًا إلى تحسين الأداء بشكل أفضل من تحسين الحالة النادرة. ومن المفارقات، أن الحالة الشائعة غالبًا ما تكون أبسط من الحالة النادرة وبالتالي يسهل تحسينها غالبًا. تشير نصيحة الفطرة السليمة إلى أنك تعرف الحالة الشائعة، والتي لا يمكن تحقيقها إلا من خلال التجربة والقياس الدقيقين.



Make the Common Case

4- الأداء عبر التوازي Performance via Parallelism

منذ فجر الحوسبة، ابتكر مهندسو الكمبيوتر تصميمات غير متوقعة المزيد من الأداء من خلال إجراء عمليات بالتوازي.



Performance via Parallelism

5- الأداء عبر خط الأنابيب Performance via Pipelining

ينتشر نمط معين من التوازي في هندسة الكمبيوتر لدرجة أنه يستحق اسمه: خط الأنابيب. على سبيل المثال، قبل سيارات الإطفاء، يستجيب "لواء دلو" للحريق، والذي تعرضه العديد من أفلام رعاة البقر ردًا على فعل غادر قام به الشرير. يشكل سكان المدينة سلسلة بشرية لنقل المياه إلى مصدر الحريق، حيث يمكنهم نقل الدلاء بسرعة أكبر إلى أعلى السلسلة بدلاً من تشغيل الأفراد ذهابًا وإيابًا لإطفاء الحريق.



Performance via Pipelining

6- الأداء عن طريق التنبؤ Performance via Prediction

في بعض الحالات، قد يكون من الأسرع في المتوسط التخمين وبدء العمل بدلاً من الانتظار حتى تعرف على وجه اليقين، بافتراض أن آلية التعافي من سوء التوقع ليست باهظة الثمن وأن تنبؤك دقيق نسبيًا.



Performance via Prediction

7- التسلسل الهرمي للذكريات Hierarchy of Memories



Hierarchy of Memories

يريد المبرمجون أن تكون الذاكرة سريعة وكبيرة ورخيصة، لأن سرعة الذاكرة غالبًا ما تشكل الأداء، وتحد السعة من حجم المشكلات التي يمكن حلها، وغالبًا ما تكون تكلفة الذاكرة اليوم هي غالبية تكلفة الكمبيوتر. لقد وجد المهندسون المعماريون أنه يمكنهم معالجة هذه الطلبات المتضاربة من خلال تسلسل هرمي للذكريات، مع وجود أسرع وأصغر وأعلى ذاكرة لكل بت في أعلى التسلسل الهرمي والأبط والأكبر والأرخص لكل بت في الأسفل. نستخدم أيقونة المثلث متعدد الطبقات لتمثيل التسلسل الهرمي للذاكرة. يشير الشكل إلى السرعة والتكلفة والحجم: كلما اقتربنا من القمة، كانت الذاكرة أسرع وأكثر تكلفة لكل بت؛ كلما اتسعت قاعدة الطبقة، زادت الذاكرة.

8- الموثوقية عبر التكرار Dependability via Redundancy



Dependability via Redundancy

لا يجب أن تكون أجهزة الكمبيوتر سريعة فحسب؛ يجب أن يكونوا موثوقين. نظرًا لأن أي جهاز مادي يمكن أن يفشل، فإننا نجعل الأنظمة يمكن الاعتماد عليها من خلال تضمين المكونات الزائدة التي يمكن أن تتولى المسؤولية عند حدوث عطل وللمساعدة في اكتشاف الأعطال.

المرجع:

Patterson, D. A., & Hennessey, J. L. (2016). *Computer organization and design ARM edition: the hardware software interface*. Morgan kaufmann.