

Communication Centric Computer Design

Simon Moore
Computer Architecture Group
Computer Laboratory
University of Cambridge

1971



2007

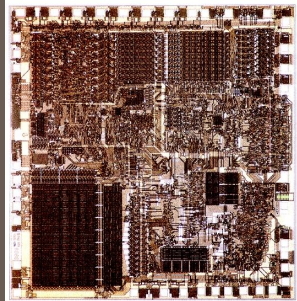


*It's the heat and the thickness of the mat
which matters.*

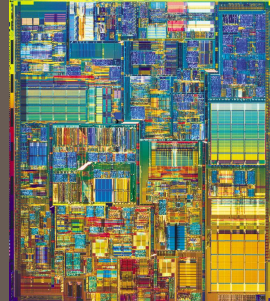
Seymour Cray
1970s



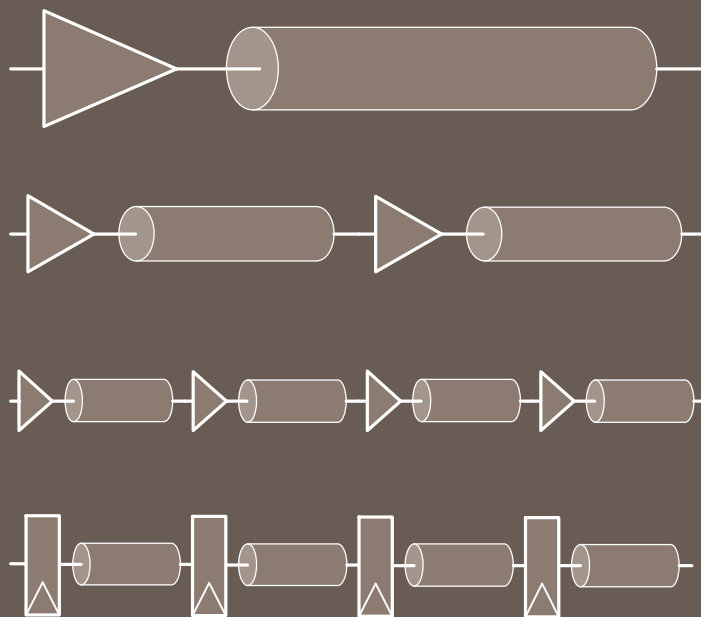
Problem: across chip wire lengths
stopped scaling



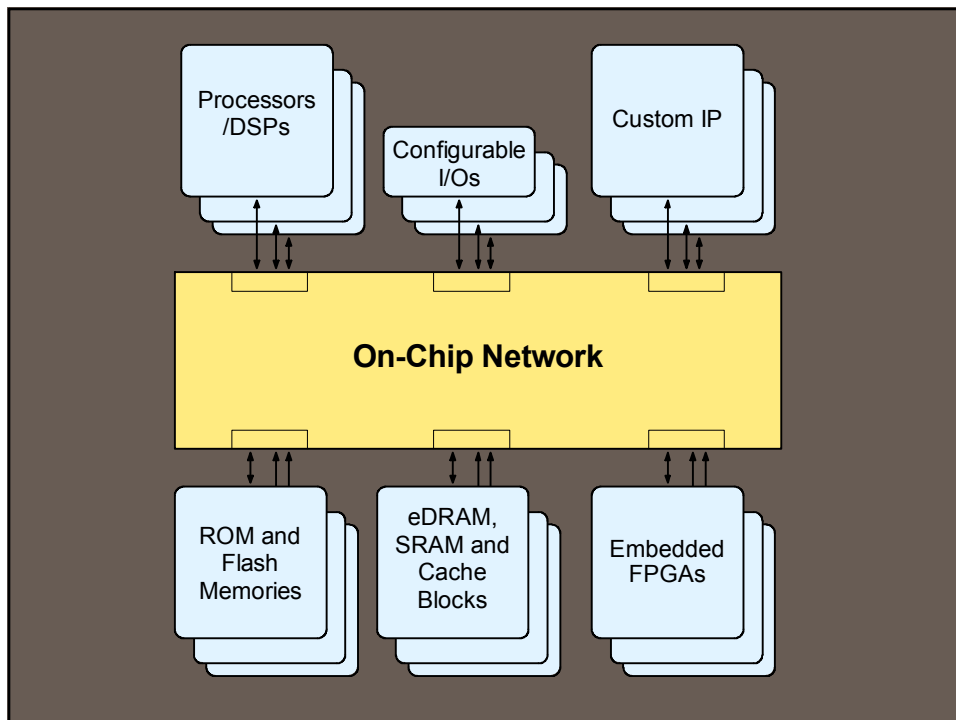
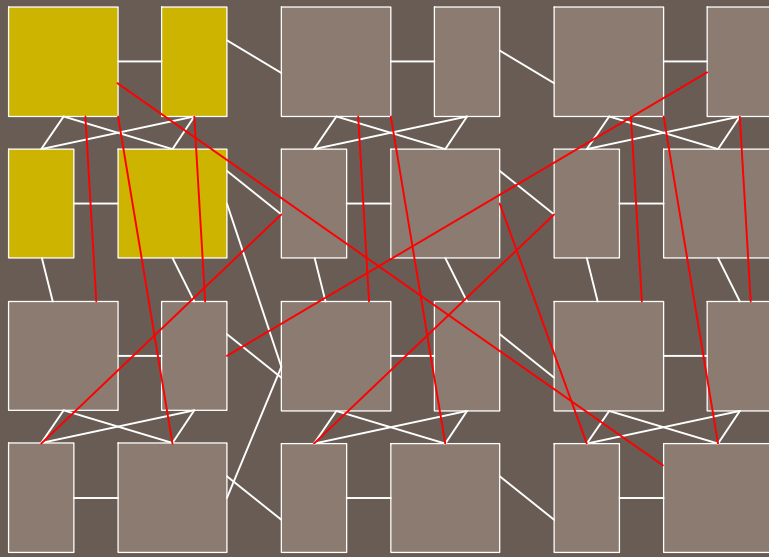
Intel 8086 (1978)
29,000 transistors
3 μ m process

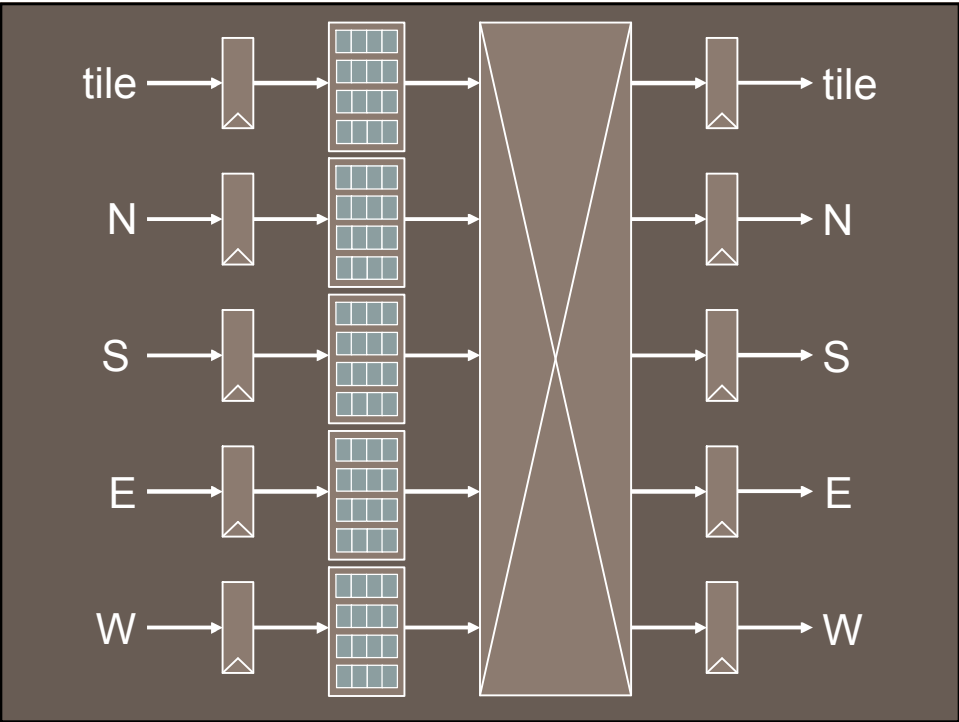
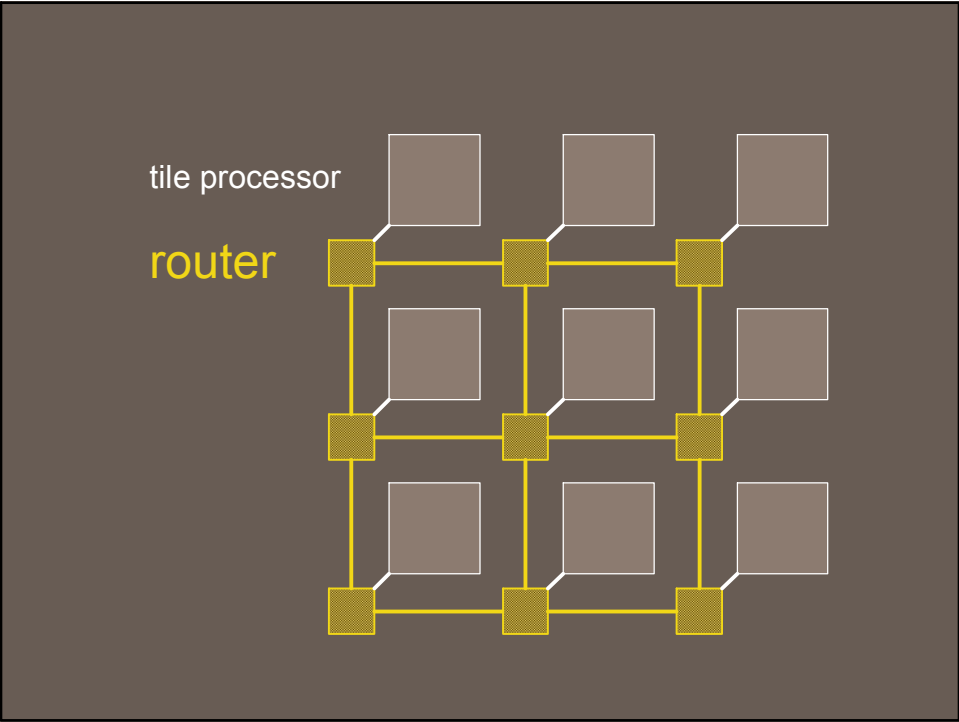


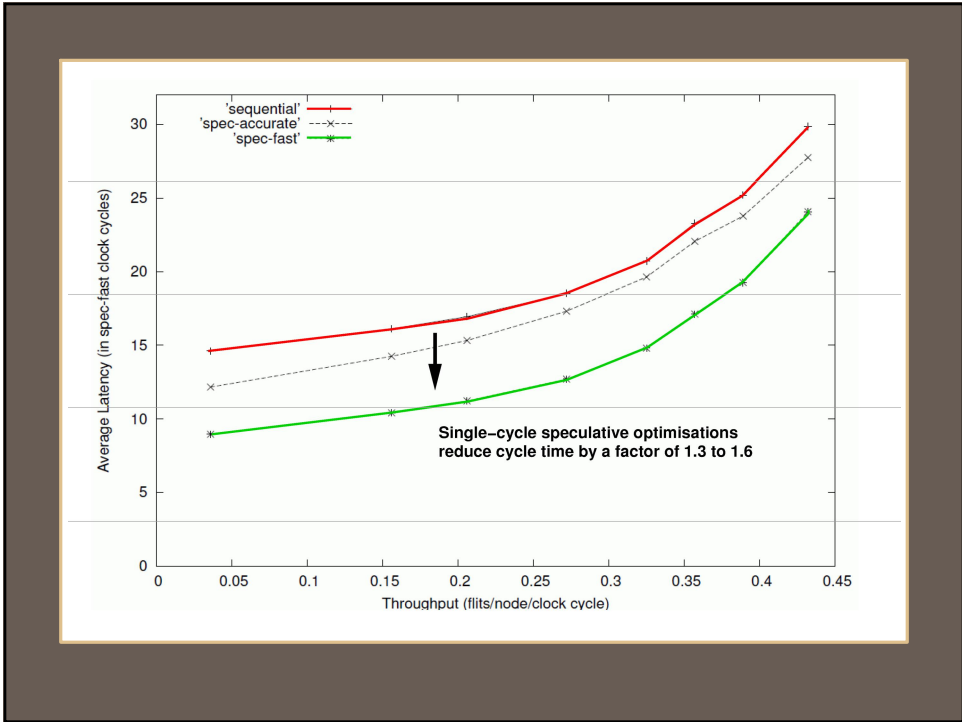
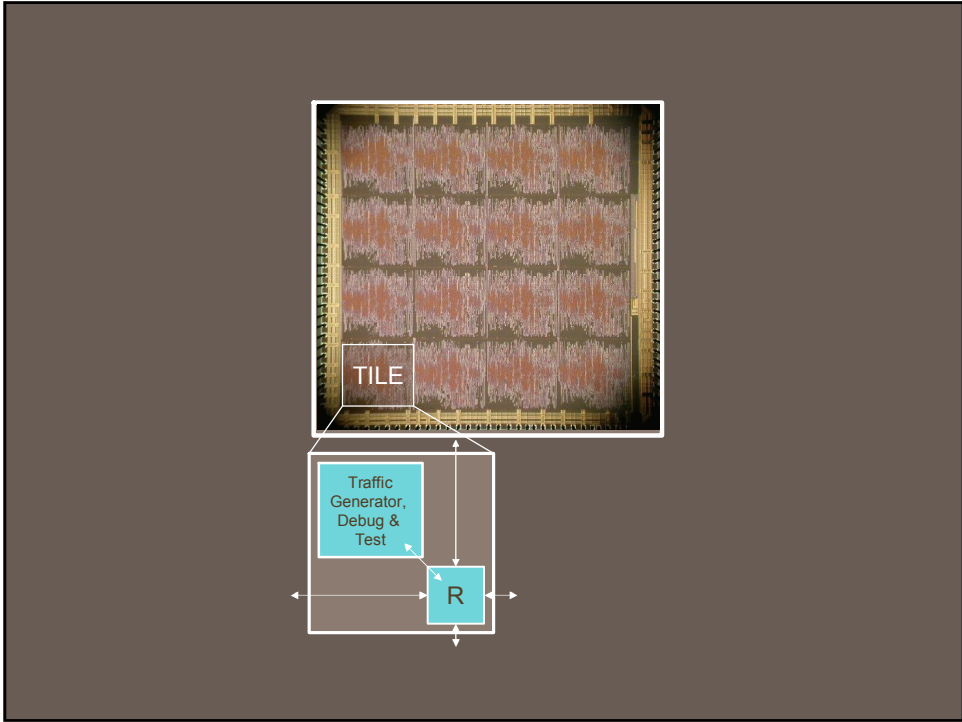
Intel Pentium-4 (2000)
42,000,000 transistors
0.18 μ m process



Rent's Rule







Computer Architecture

technology scaling favours
transistors over wires

Power consumed by Computation vs. Communication

technology node	130nm CMOS	50nm CMOS
transfer 32b across chip	20 ALU ops	57 ALU ops
transfer 32b off-chip	260 ALU ops	1333 ALU ops

communication rather than
computation **limits performance**

When did computer
architecture stop scaling?

Classic Scaling

With every feature size scaling of n

- you get $O(n^2)$ transistors
- they run $O(n)$ faster

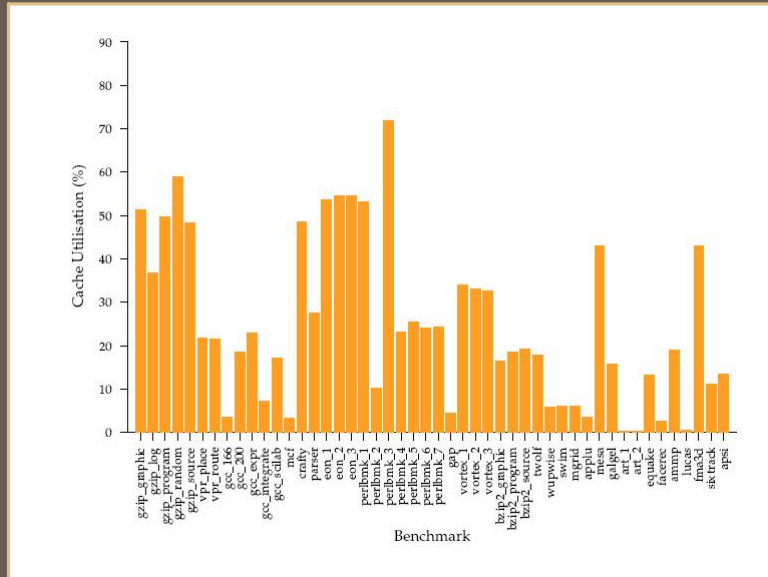
Processor scaling eras (Jouppi)

- n^3 era
 - n from device speed, n^2 from transistor count
- n^2 era
 - n from device speed, n from n^2 transistors
- n era
 - n from device speed, 1 from n^2 transistors

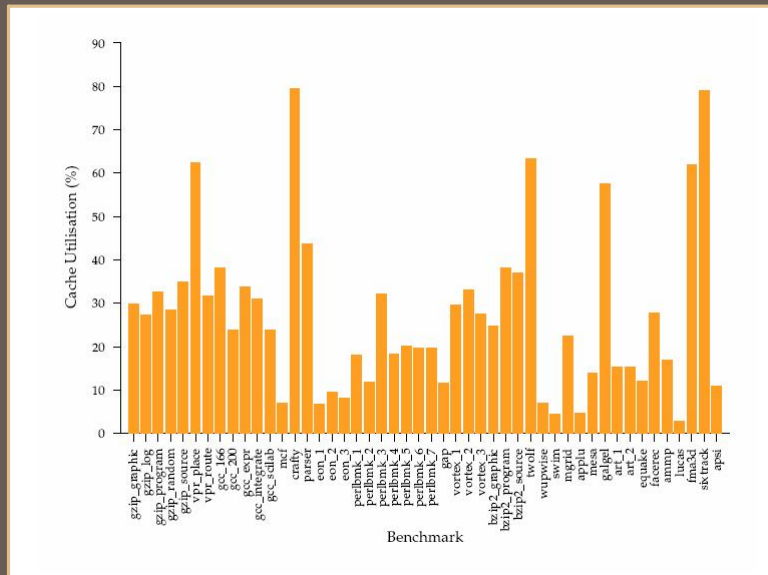
Computation: the case for many simpler cores is easy to make

Communication: but what about memory hierarchy and communication between threads?

Level-1 D-cache utilisation



Level-2 unified cache utilisation



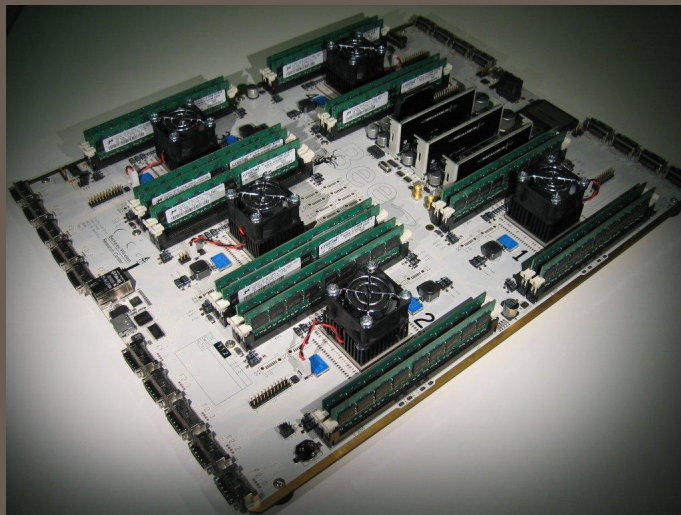
often 80% of the cache holds dead data

Communication Challenges

1. How do we get the data to the processor or the processing to the data?
2. How should threads communicate?

Computer Architecture Sandpit

FPGAs for architecture **simulation**



Example FPGA board from Berkeley

Conclusions

- Parallel processing for the masses is finally here, but how do we use such resources?
- Communication complexity prompts new computer architecture
- FPGAs are now big enough & have enough connectivity to allow parallel systems to be easily and efficiently simulated



Contact Details

Dr Simon Moore

Computer Laboratory
University of Cambridge
JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom

E-mail: simon.moore@cl.cam.ac.uk

URL: <http://www.cl.cam.ac.uk/users/swm11>