



# NTNU

Det skapende universitet

## **Reconfigurable Computing**

Kenneth Østby <kenneo@idi.ntnu.no>

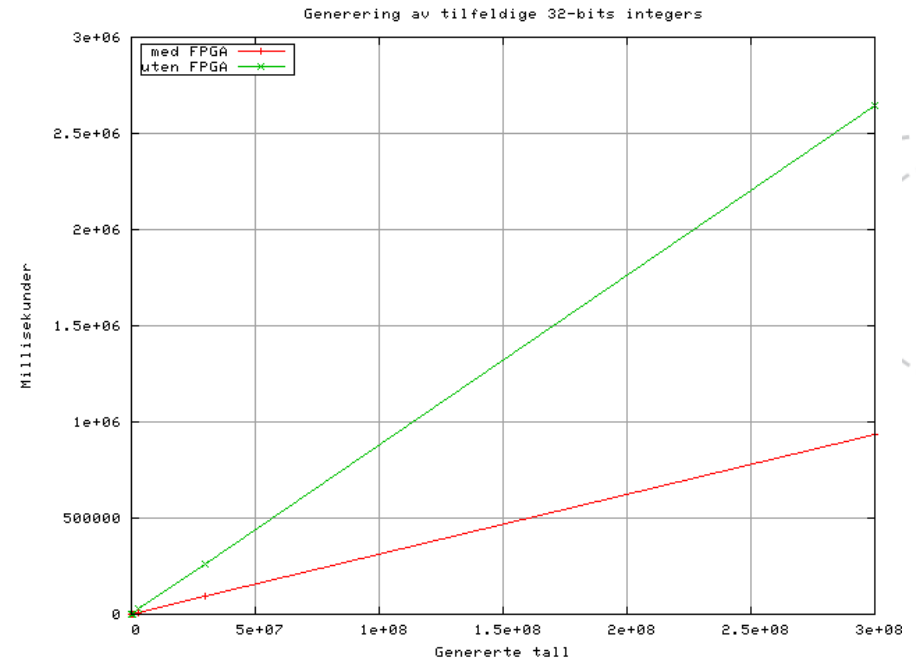
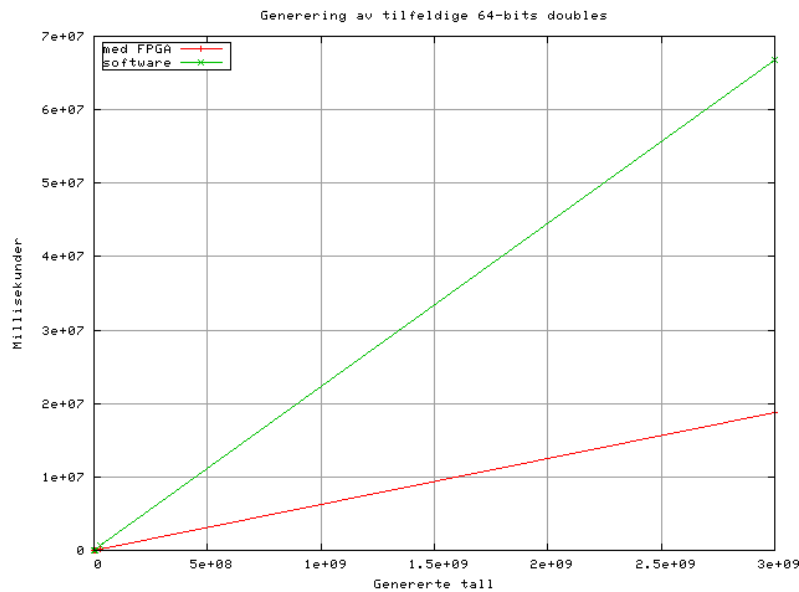
# Traditional Methods

- ASIC
  - High Performance
  - Behaviour can't be altered after fabrication
- Programmable microcontroller
  - Flexible
  - Overhead in work load, decreases performance

# FPGA Speedup

- Previously used mainly for prototyping.
- Higher performance at some tasks:
  - Serpent Block Cipher ( factor 18 )
  - Implementation of sieving for factoring large numbers ( factor 28 )
    - Useful for breaking encryption
  - DES and Elliptic curve crypto
  - String pattern matching
  - Genetic Algorithms for the travelling salesman problem

# Speedup from musculus (Cray XD1)



Mersenne Twister 19937

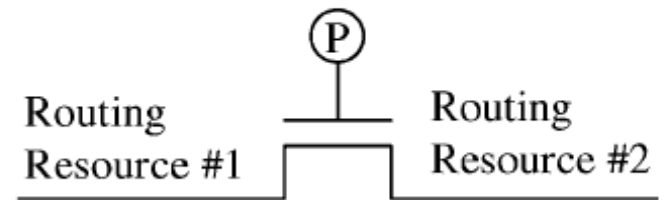
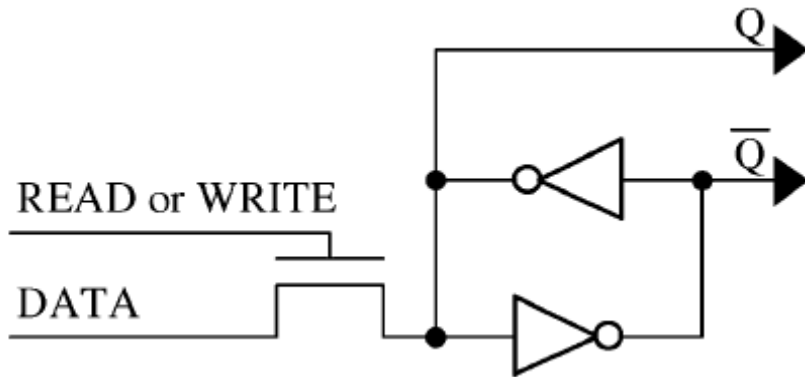
# Motivation

- Some tasks are best solved using Hardware
  - Computational Logic
- Some tasks are best solved using Software
  - Data-dependant control
  - Memory Access
- *“Yes please, both” -- Winnie the Pooh*

# Reconfigurable Computing

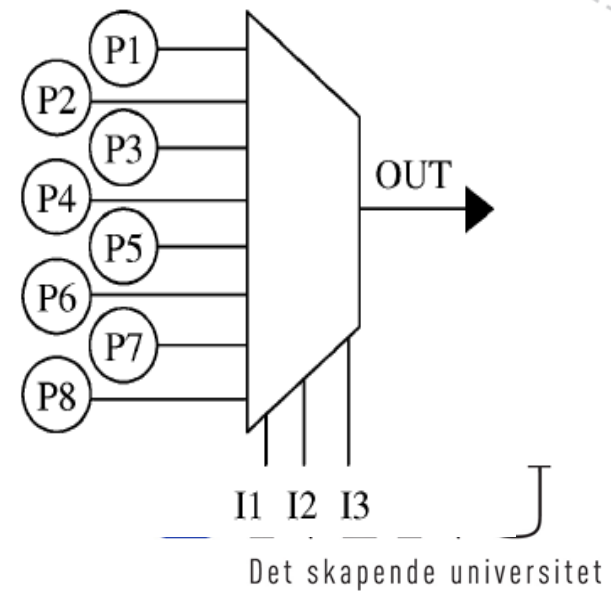
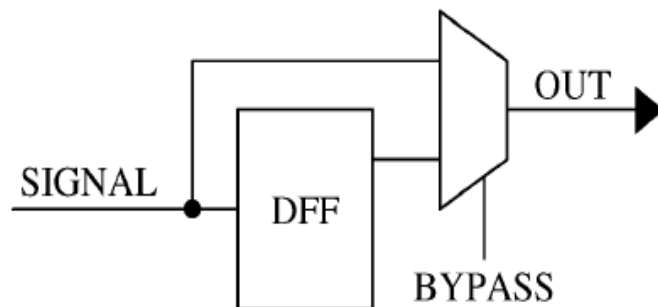
# FPGA Technology

- SRAM-programmable
- Programming the SRAM-bits configures the FPGA.



# More control bits

- Lookup Table ( Best
- D flip-flop
- ALU ( Decide operation: add, subtract, divide, multiply

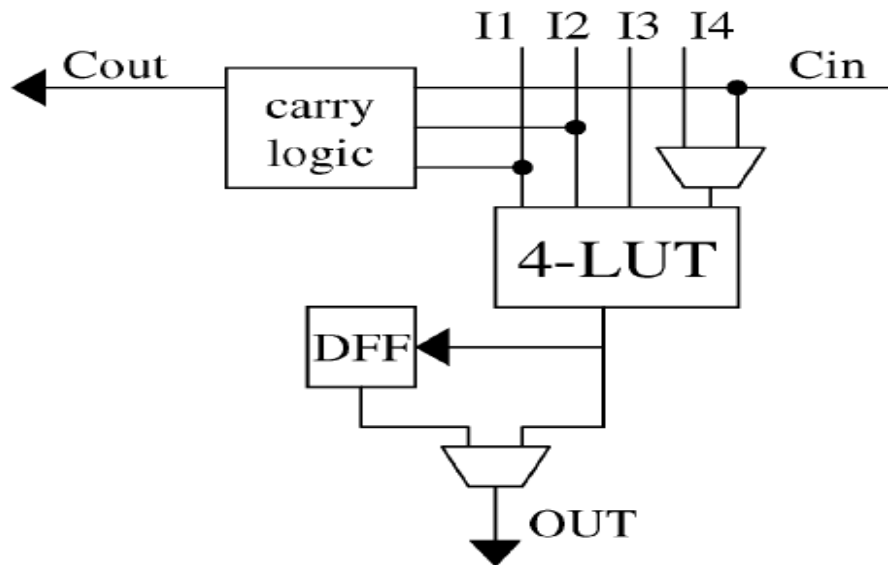


# Granularity

- Finer granularity allows for better control
- Coarse granularity for specialized tasks
- On some Xilinx they embed multipliers and divisors which are coarse grained units
- LUTS are fine grained.

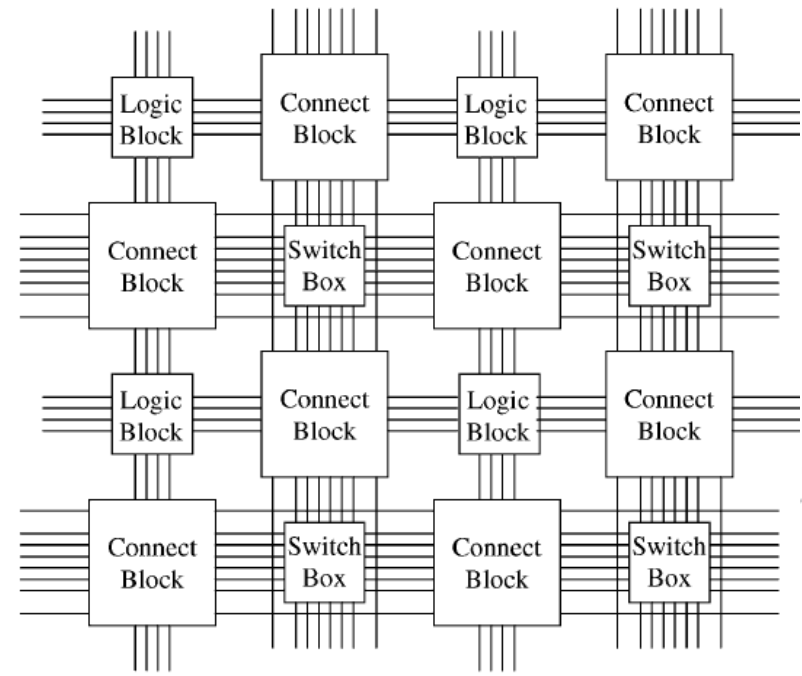
# Logic block

- Used to implement generic logic
- Basic building block inside the FPGA
- Route several together for desired effect



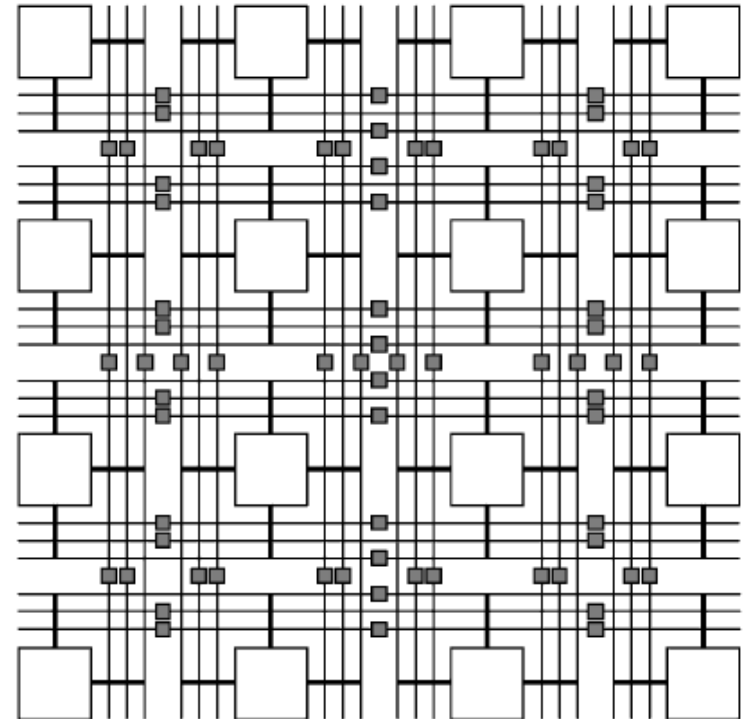
# Routing: Island Style

- 90% areal of chip used at routing
- Logic block connected to Connect Block
- Different length on wires
  - Local lines, ie: carry chains
  - Medium lines: several logic blocks
  - Long lines: Over entire chip
  - Global lines: Clock signal and such.



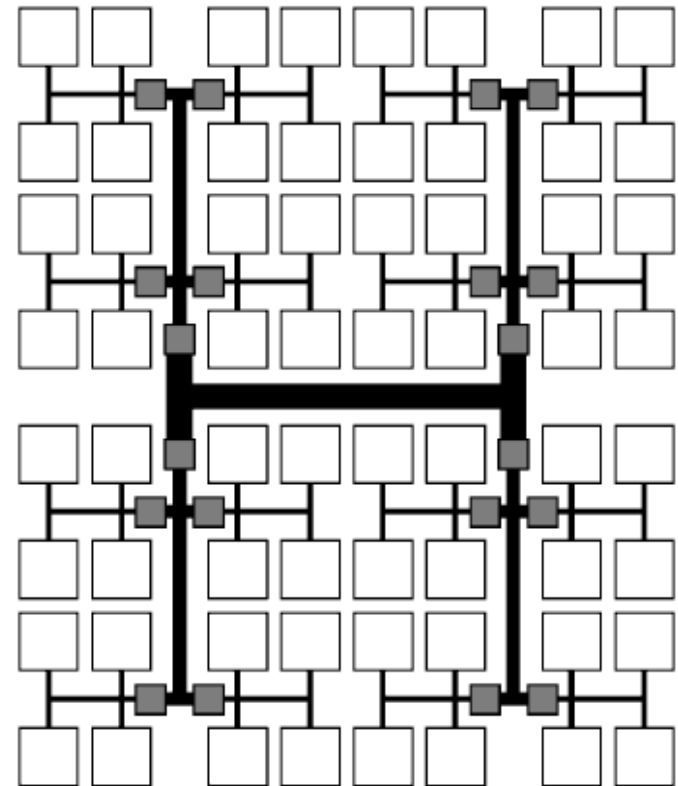
# Segmented Routing

- Connect several short wires
- Have some longer wires



# Hierarchical Routing

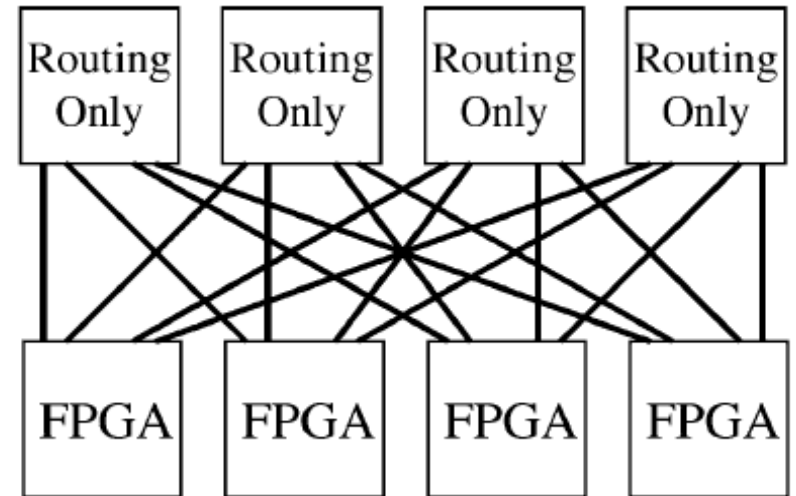
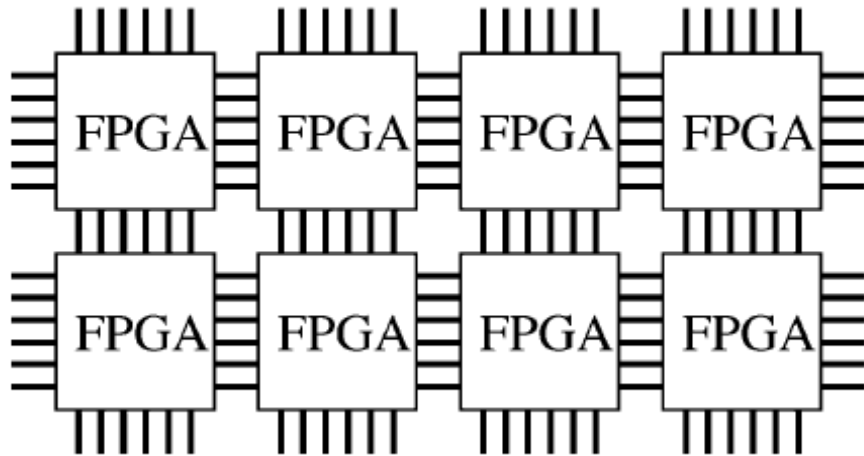
- Small and quick lines between local logic
- Connect logic groups with bigger wires
- Connect those with even bigger wires



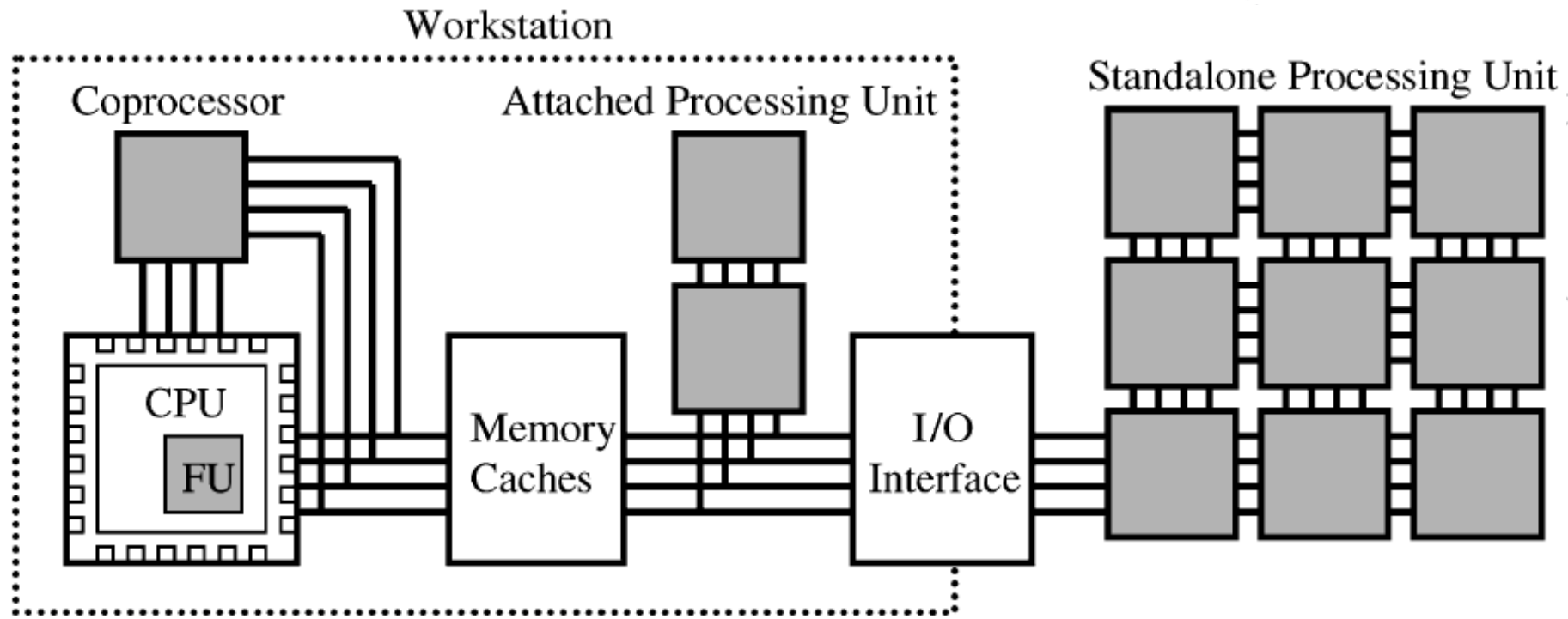
# Routing demands an Engineering decision!

# Multi-FPGA system

- When the design cannot be partitioned over 1 FPGA
- Mesh – Unefficient if talking to FPGAs with greater manhattan distance of 1
- Crossbar – Major overhead

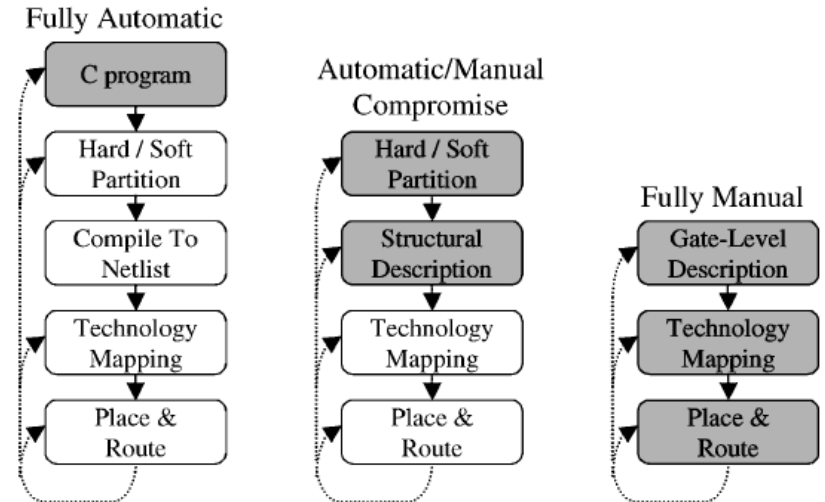


# Connected



# Software - Designflow

- Fully Manual gives powerful control over the system, but increases the level of complexity
- Fully automatic easier for the programmer, might not be as efficient



# Partitioning Hardware/Software

- Complex control sequences in software
- Fixed datapaths in hardware
- Automatic by pricing the functions
- Semi-automatic by declaring which functions to be driven by HW ( NAPA C )
- Do everything yourself using VI

# From HLL => FPGA

- 1) Write in High Level Language ( Java, C++ )
- 2) Generate structural description
- 3) Break down complex structures into elemental gates
- 4) Map into the technological elements to the FPGA  
( Technology Mapping )
- 5) Perform Floorplanning, place the structures on chip
- 6) Route between elements. Depends on a good #5

Reconfigurable Computing  
requires Engineering  
decisions!