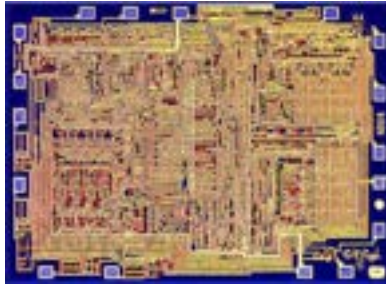
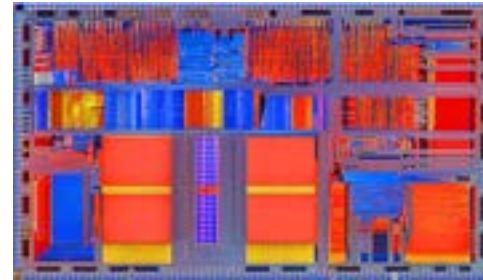


Intel i4004 (1971)



2,300 transistors

Intel i486 (1988)

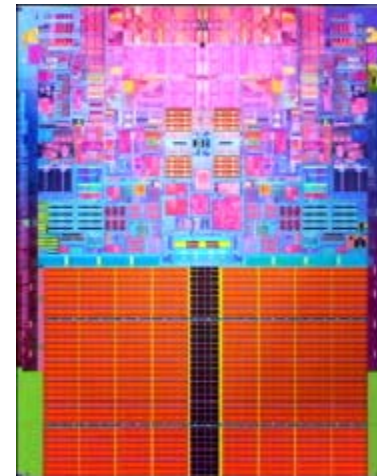


1.2 million transistors



P4 (2000)

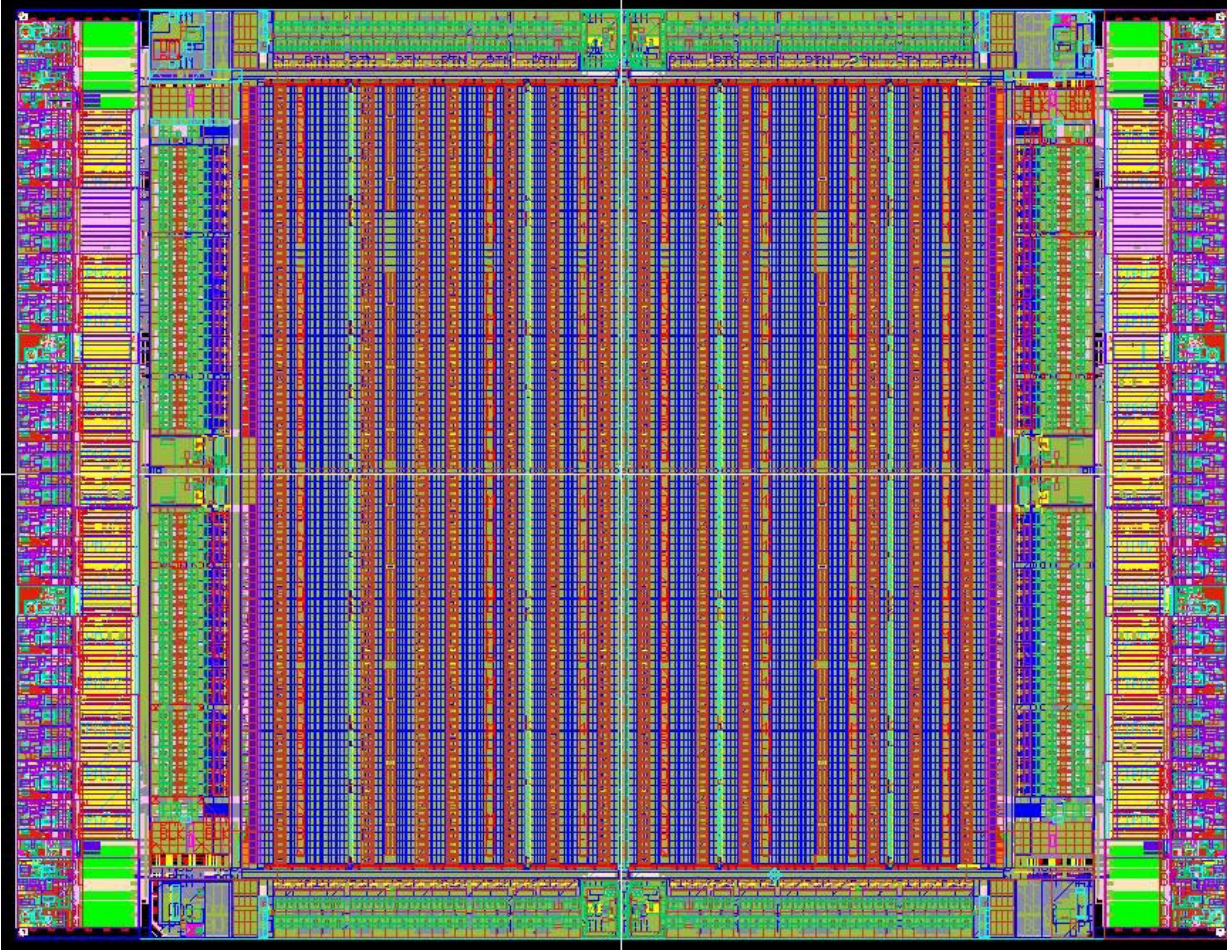
55 million transistors



**Core2 Duo
(2006)**

291 million transistors

Altera Stratix V FPGA (2011)



**3.9 billion
transistors**

© 2011 Altera Corporation

ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS & STRATIX are Reg. U.S. Pat. & Tm. Off.
and Altera marks in and outside the U.S.



What are FPGAs and why should we use them?

FPGA

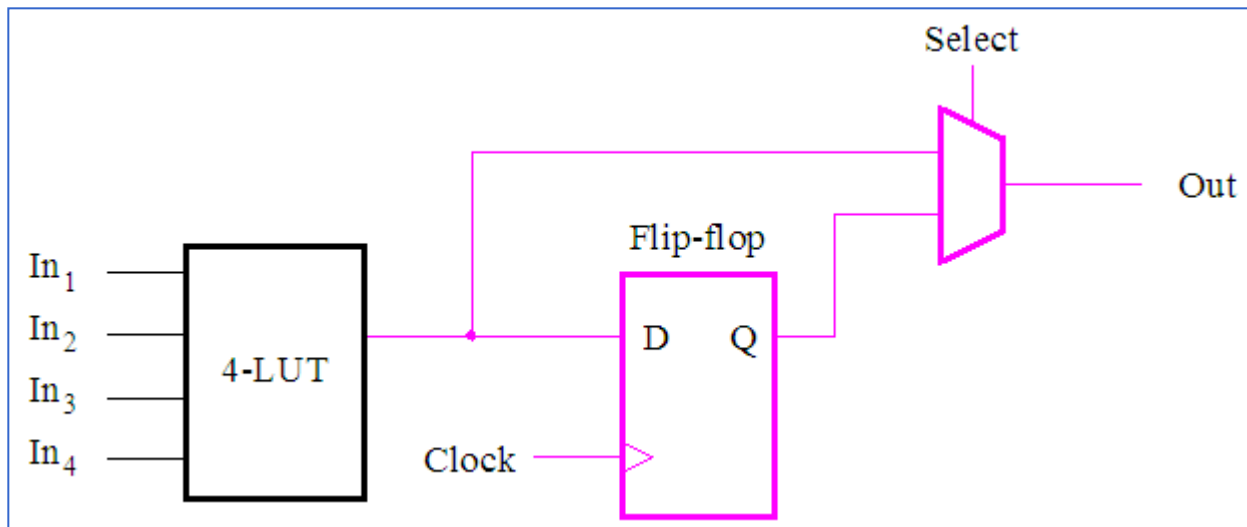
■ **Field Programmable** Gate Array

- Two-dimensional array of *programmable* logic elements
- Connected with *programmable switches*

■ FPGA: re-programmable hardware

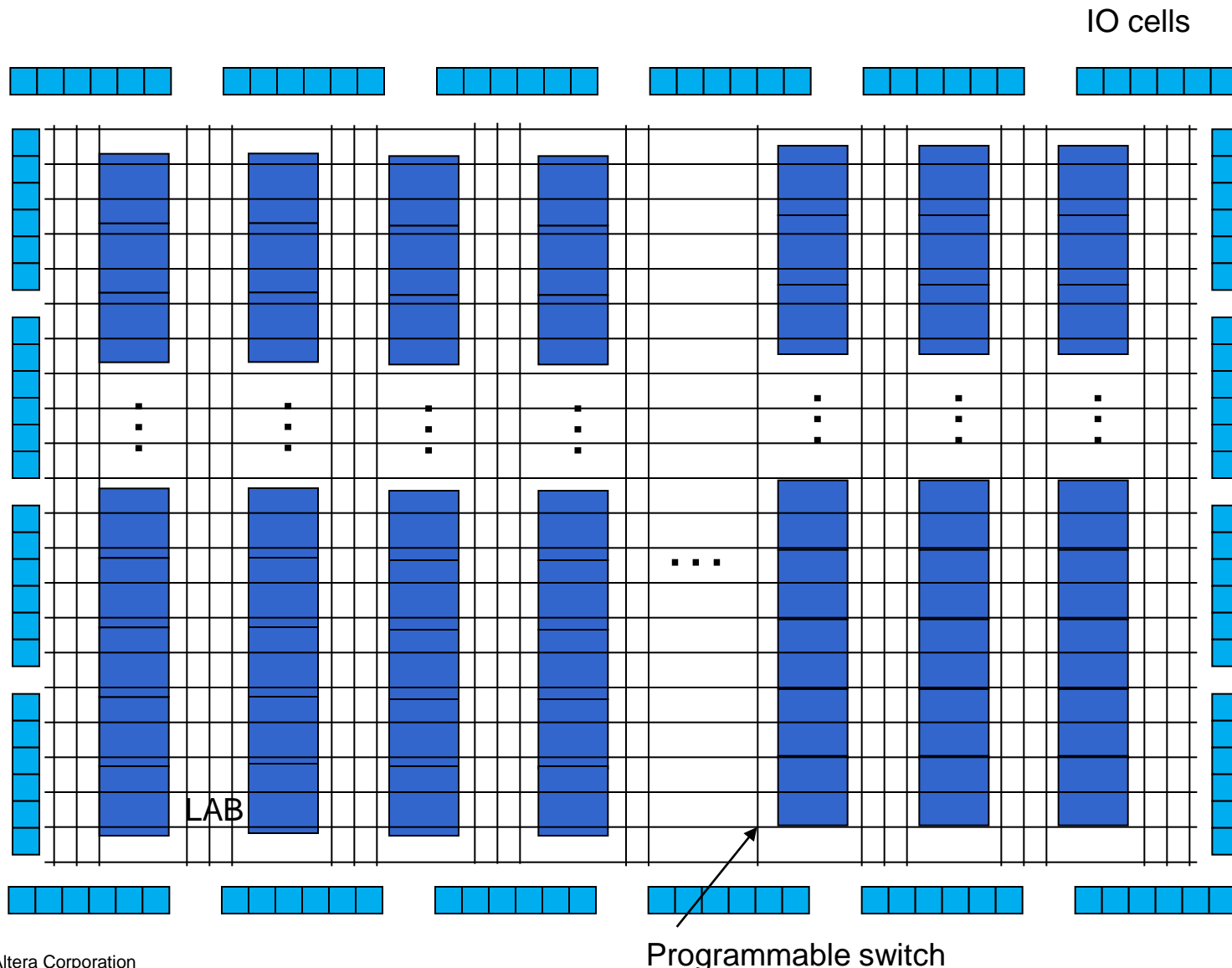
FPGA Architecture: Logic Element

- Lookup table (LUT) implements any 4-input logic function



- Actual LE is significantly more complex

FPGA Architecture

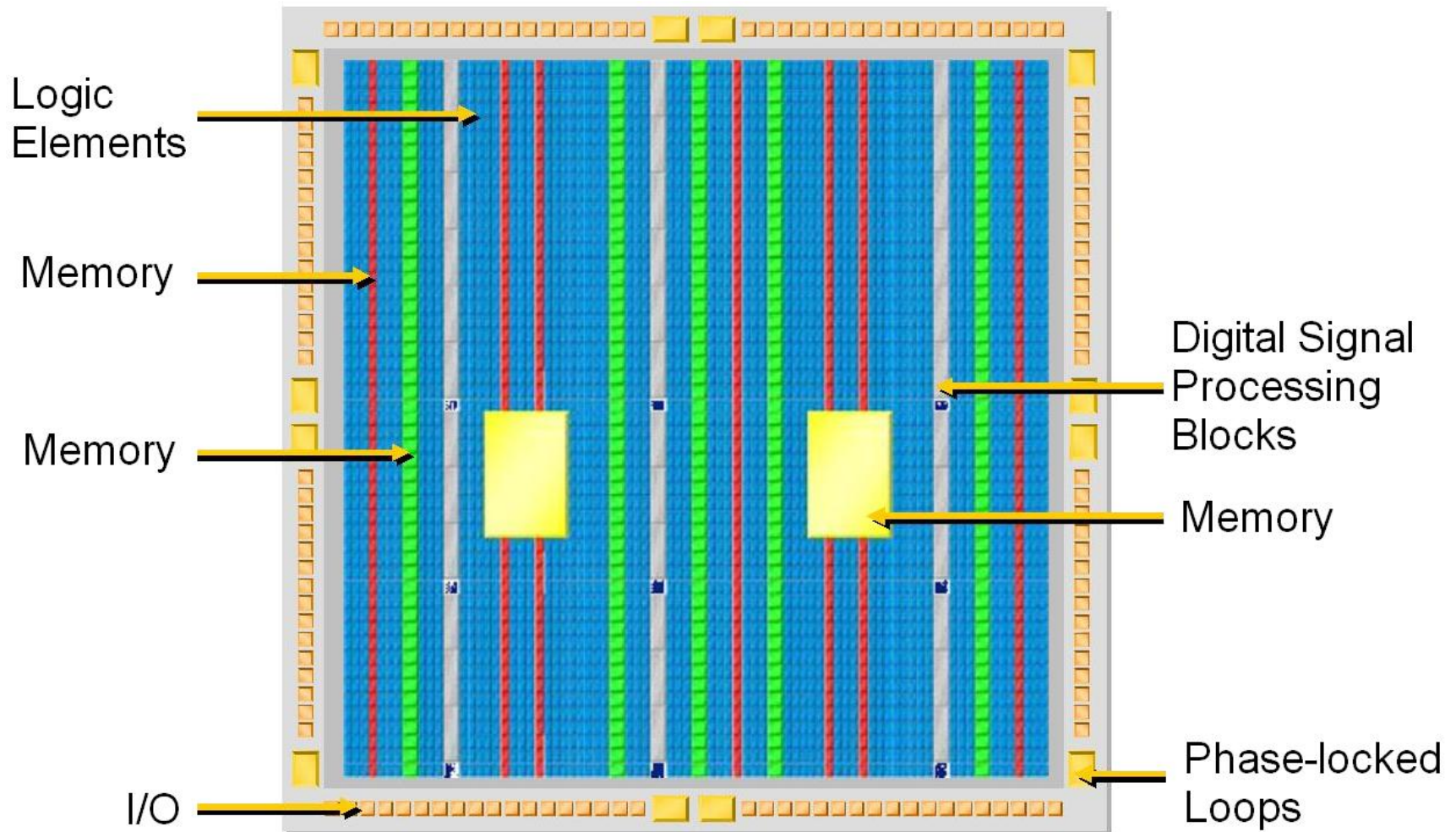


© 2011 Altera Corporation

ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS & STRATIX are Reg. U.S. Pat. & Tm. Off.
and Altera marks in and outside the U.S.

ALTERA

Example: Altera Stratix-series FPGA



Why use FPGAs?

■ Problem:

- Create an application to process a lot of data quickly
- How?

■ Alternatives:

- Processor?
 - **Easy** to write code, **low performance**, **power** hungry
- Gate Array (ASIC)?
 - Very high **performance**, **low power**, very **hard** to design, **expensive** to manufacture
- Field-Programmable Gate Array?
 - **no manufacturing** needed (just program), **easier** to design for than ASIC, high **performance**, **lower power** than a processor