FPGA and ASIC

How to implement a digital system

No two applications are identical and every one needs certain amount of customization

Basic methods for customization

a) “General-purpose hardware” with custom software
   • General purpose processor: e.g., performance-oriented processor (e.g., Pentium), cost-oriented processor (e.g., PIC micro-controller)
   • Special purpose processor: architecture with a specific set of functions: e.g., DSP processor (to do multiplication-addition), network processor (to do buffering and routing), “graphic engine” (to do 3D rendering)

b) Custom hardware (no software)

c) Custom software on a custom platform (CPU+other hardware), known as hardware-software co-design
How to implement a digital system (2)

Trade-off between flexibility, programmability, design effort, cost, performance, and power consumption

DEVICE TECHNOLOGIES
What does an IC look like?

Several metal layers
  – Less congestion

Hierarchical scaling

Wires on top levels are wider and taller than on lower levels

Top layers are for
  – Power supply
  – Clock
  – Global signals

What does an IC look like? (2)

Intel dual core
What does an IC look like? (3)

45 nm, quad-core
Note the symmetry
Two dual-cores integrated

Actel Fusion Mixed-signal FPGA
1. Integrated Analog-to-Digital Converter (ADC)
2. Fusion Supports Low Power, synchronization
3. Embedded Flash Memory
4. Advanced I/O Standards
5. Charge Pumps
6. Analog Quads
7. Flash FPGA VersaTile
8. SRAM and FIFOs
9. Integrated Oscillators—Crystal and RC
10. Routing Structure
11. JTAG
Classification of device technologies

Where customization is done:
- In the *fab* (fabrication facility): ASIC (Application Specific IC)
  - Full-custom ASIC
  - Standard cell ASIC
  - Gate array ASIC
    - Lower density and performance than other ASICs, but more expensive than non-ASIC => obsolete
- In the “field”: non-ASIC
  - Complex field-programmable logic device
  - Simple field-programmable logic device
    - Replaced by CPLD/FPGA
  - Off-the-shelf SSI (Small Scaled IC)/MSI (Medium Scaled IC) components
    - No longer a viable option!

ASIC

Full-custom
- All aspects (e.g., size of a transistor) of a circuit are tailored for a particular application.
- Circuit fully optimized
- Design extremely complex
- Very time consuming design (typically only feasible for small components)
- Intel and AMD are partly full-custom

Standard-cell
- Circuit made of a set of pre-defined logic, known as standard cells
- Layout of a cell is pre-determined, but layout of the complete circuit is customized
- Eg. Mobile phone digital ICs
Complex Field Programmable Logic Device

Device consists of an array of generic logic cells and general interconnect structure.

Logic cells and interconnect can be “programmed” by utilizing “semiconductor fuses” or “switches.”

Customization is done “in the field.”

Two categories:

- CPLD (Complex Programmable Logic Device)
  - sea-of-gates to implement logic
- FPGA (Field Programmable Gate Array)
  - Look-up tables to implement logic

No custom mask needed

For example, Cisco 2600 series routers

---

Comparison of technology

Area (Size): silicon real-estate: [mm²], [eq. gates]

Speed (Performance): [MHz], [op/s]
  - Operations/second
  - i.e. Time required to perform a task

Power consumption [mW]

Cost [€]

Design effort [person-month]
Comparison of technology

Area: ASIC (sc) vs FPGA
- Standard cell is the smallest since the cells and interconnect are customized
- FPGA is the largest
  - Overhead for programmability
  - Capacity cannot be completely utilized
- Roughly: FPGA is approximately 35 times larger using the LUT-based logic elements [1]
  - However, that is not seen by FPGA end users – high volume compensates some costs ($$)

Performance: ASIC (sc) vs FPGA
- Roughly: FPGA is between 3.4 to 4.6 times slower, MHz [1]


Cost

Types of cost:
- Chip design costs
  - NRE (Non-Recurring Engineering) cost: one-time, per-design cost
  - Part cost: per-unit cost
- Indirect design costs
  - Lead time: time to get the chip out of the factory
  - Time-to-market "cost" loss of revenue

Standard cell: high NRE, small part cost and large lead time
- Good for large volumes

FPGA: low NRE, large part cost and small lead time
- Good for prototypes and small volumes
## Summary of technologies

Trade-off between optimal use of hardware resource and design effort/cost

No single best technology

<table>
<thead>
<tr>
<th></th>
<th>FPGA</th>
<th>ASIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tailored Masks</td>
<td>15 or more</td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>Best (smallest)</td>
<td></td>
</tr>
<tr>
<td>Speed</td>
<td>Best (fastest)</td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td>Best (minimal)</td>
<td></td>
</tr>
<tr>
<td>NRE Cost</td>
<td>Best (smallest)</td>
<td></td>
</tr>
<tr>
<td>Per-part Cost</td>
<td>Best (smallest)</td>
<td></td>
</tr>
<tr>
<td>Design cost</td>
<td>Best (easiest)</td>
<td></td>
</tr>
<tr>
<td>Time-to-market</td>
<td>Best (shortest)</td>
<td></td>
</tr>
</tbody>
</table>