RETI LOGICHE

Sito del corso:
http://vision.unipv.it/reti-logiche/

Sito del corso: http://vision.unipv.it/reti-logiche/

RETI LOGICHE
Corso di laurea in Ingegneria Elettronica e Informatica - a.a. 2015-2016

Prof. Virginio Cantoni

Il modulo **Reti Logiche** intende fornire i fondamenti dell'algebra di Boole, i metodi e le tecniche di analisi e di progetto delle reti logiche combinatorie e sequenziali sincrone e asincrone e una descrizione delle funzioni dell'unità aritmetica inquadrate nello scenario dell'architettura di un processore numerico.
Le esercitazioni vertono sull'analisi e sintesi di reti logiche e sugli algoritmi per le operazioni aritmetiche in presenza di un addizionatore.
Al termine del corso lo studente sarà in grado di analizzare e progettare le reti logiche più comuni e di comprendere le funzioni dell'unità aritmetica e le relative prestazioni.
Per informazioni dettagliate invitiamo a visitare la scheda del corso sul sito della Facoltà di Ingegneria.

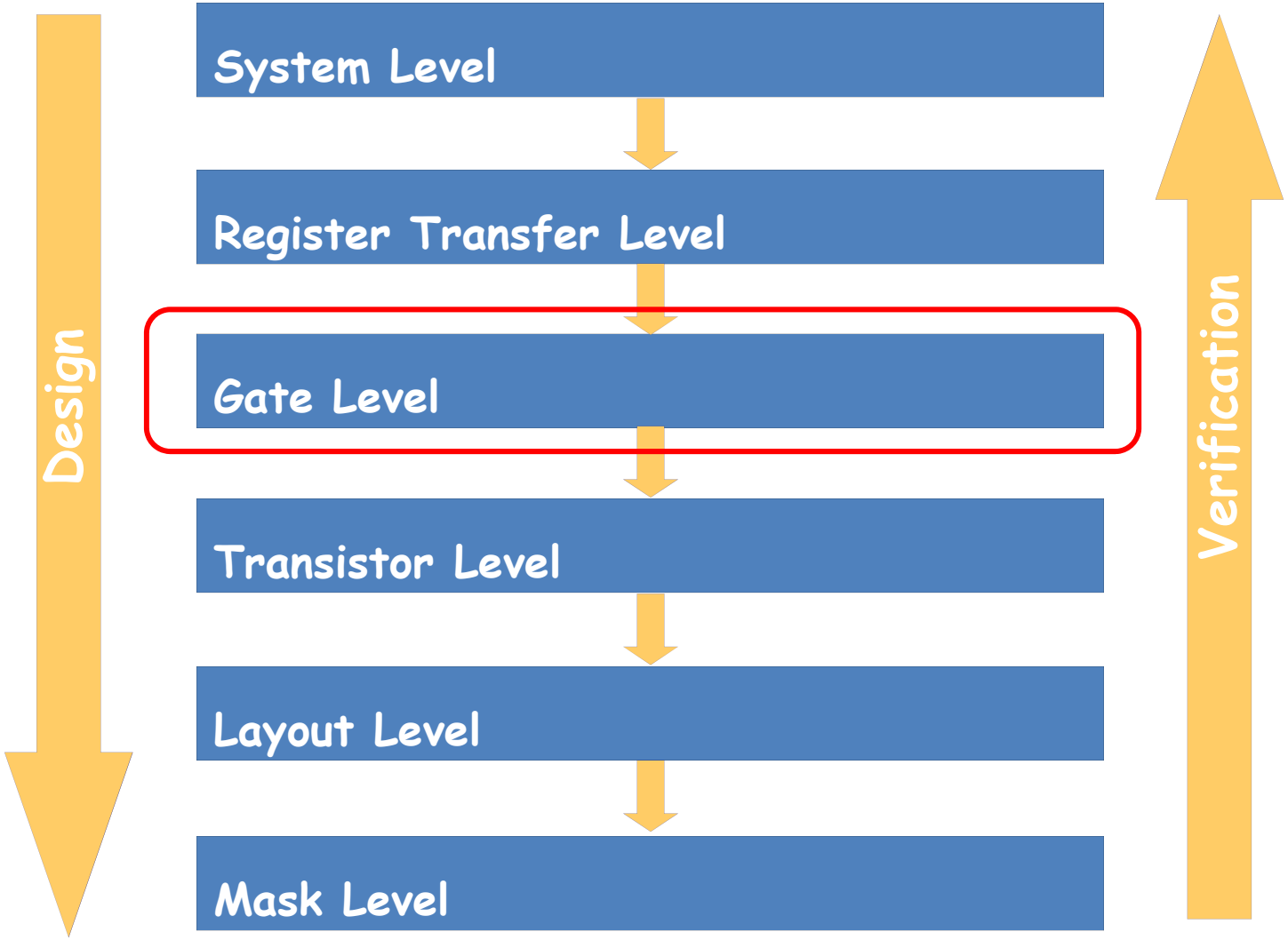
Last update, 29 February 2016

Descrizione della prova d'ESAME

ESAMI 2015/2016:

- Risultati esame PAVIA 25 febbraio 2016

Design of Integrated Digital Systems



INFORMATION REPRESENTATION - Signals

- ✓ Information variables represented by physical quantities.
- ✓ For digital systems, the variables take on discrete values.
- ✓ Two level, or binary values are the most prevalent values in digital systems.
- ✓ Binary values are represented abstractly by:
  - digits 0 and 1
  - words (symbols) False (F) and True (T)
  - words (symbols) Low (L) and High (H)
  - and words On and Off.
- ✓ Binary values are represented by values or ranges of values of physical quantities

System Level

- ✓ Abstract algorithmic description of high-level behavior
  - e.g. C-Programming language

```
Port*
compute_optimal_route_for_packet(Packet_t *packet,
                                 Channel_t *channel)
{static Queue_t *packet_queue;

  packet_queue = add_packet(packet_queue, packet);
  ...}
```

  - abstract because it does not contain any implementation details for timing or data
  - efficient to get a compact execution model as first design draft
  - difficult to maintain throughout project because no link to implementation

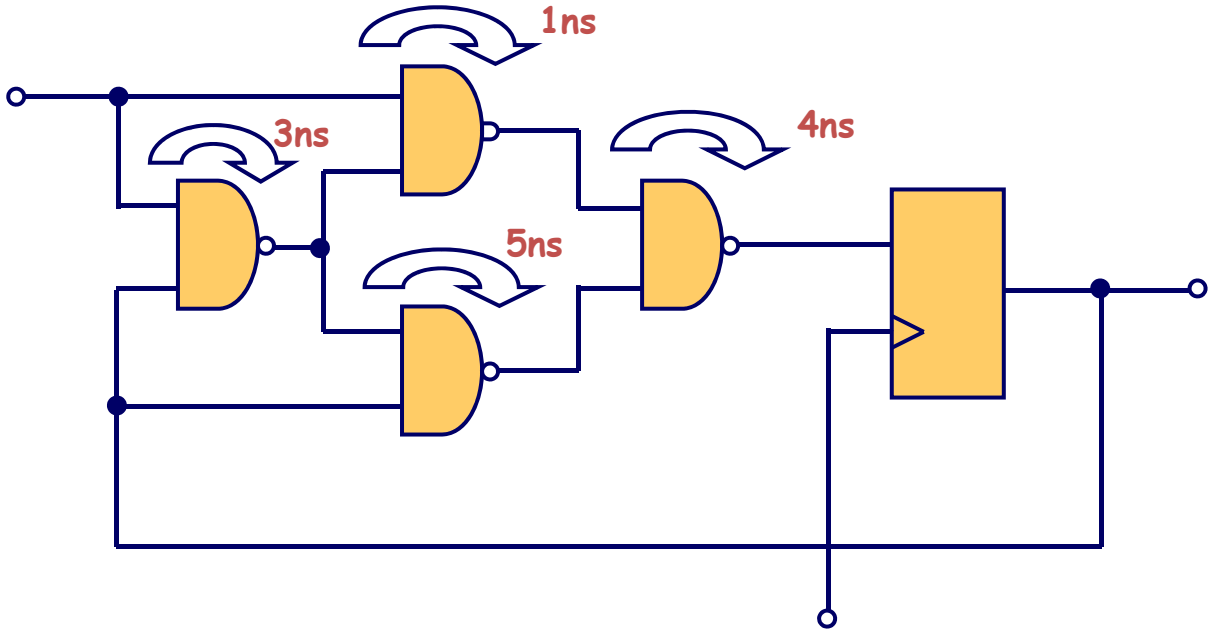
RTL Level

- ✓ Cycle accurate model "close" to the hardware implementation
  - bit-vector data types and operations as abstraction from bit-level implementation
  - sequential constructs (e.g. if - then - else, while loops) to support modeling of complex control flow

```
module mark1;
reg [31:0] m[0:8192];
reg [12:0] pc;
reg [31:0] acc;
reg[15:0] ir;
always
   begin
     ir = m[pc];
     if(ir[15:13] == 3b'000)
        pc = m[ir[12:0]];
     else if (ir[15:13] == 3'b010)
        acc = -m[ir[12:0]];
     ...
   end
endmodule
```

Gate Level

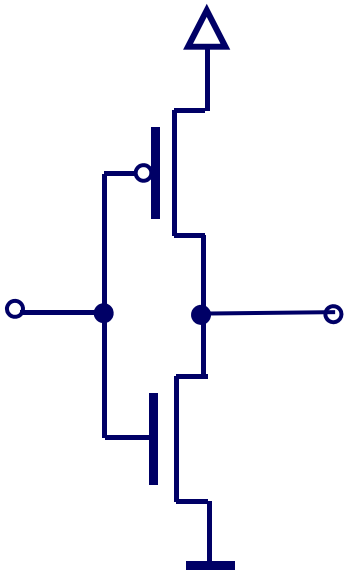
- ✓ Model on FINITE-STATE MACHINE LEVEL
  - models function in Boolean logic using registers and gates
  - various delay models for gates and wires



  - in this course we will mostly deal with gate level

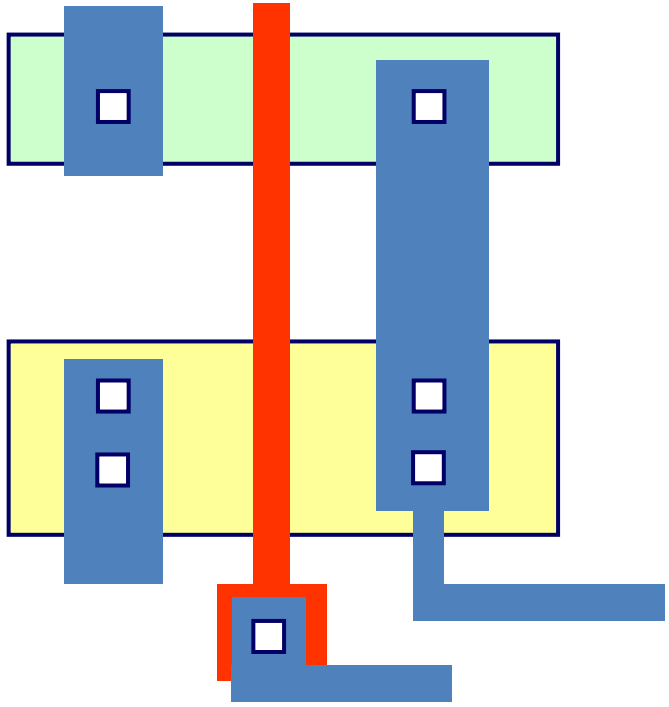
Transistor Level

- ✓ Model on CMOS transistor level
  - depending on application function modeled as resistive switches
    - used in functional checking
  - or full differential equations for circuit simulation
    - used in detailed timing analysis



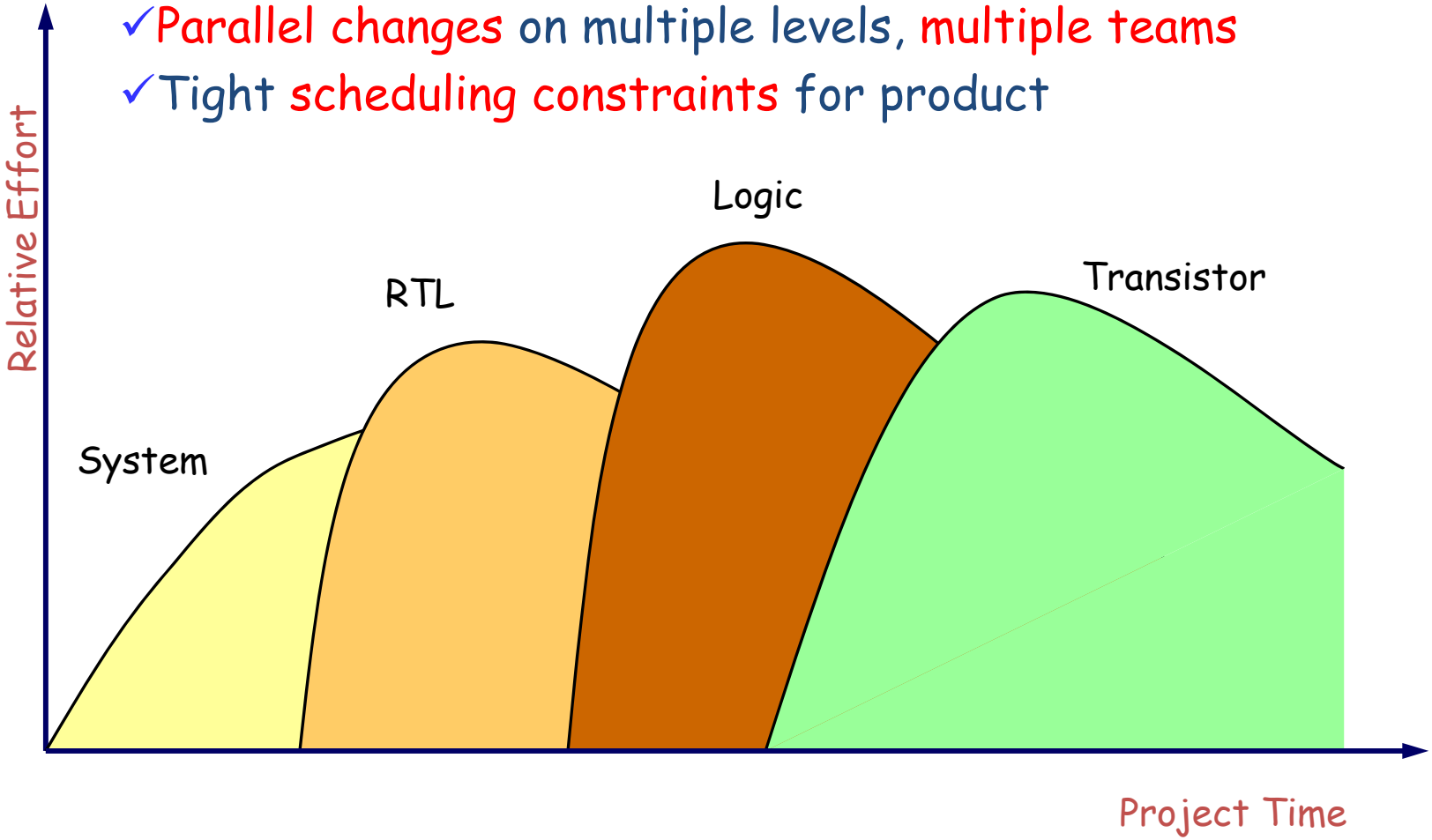
Layout Level

- ✓ Transistors and wires are laid out as polygons in different technology layers such as diffusion, poly-silicon, metal, etc.

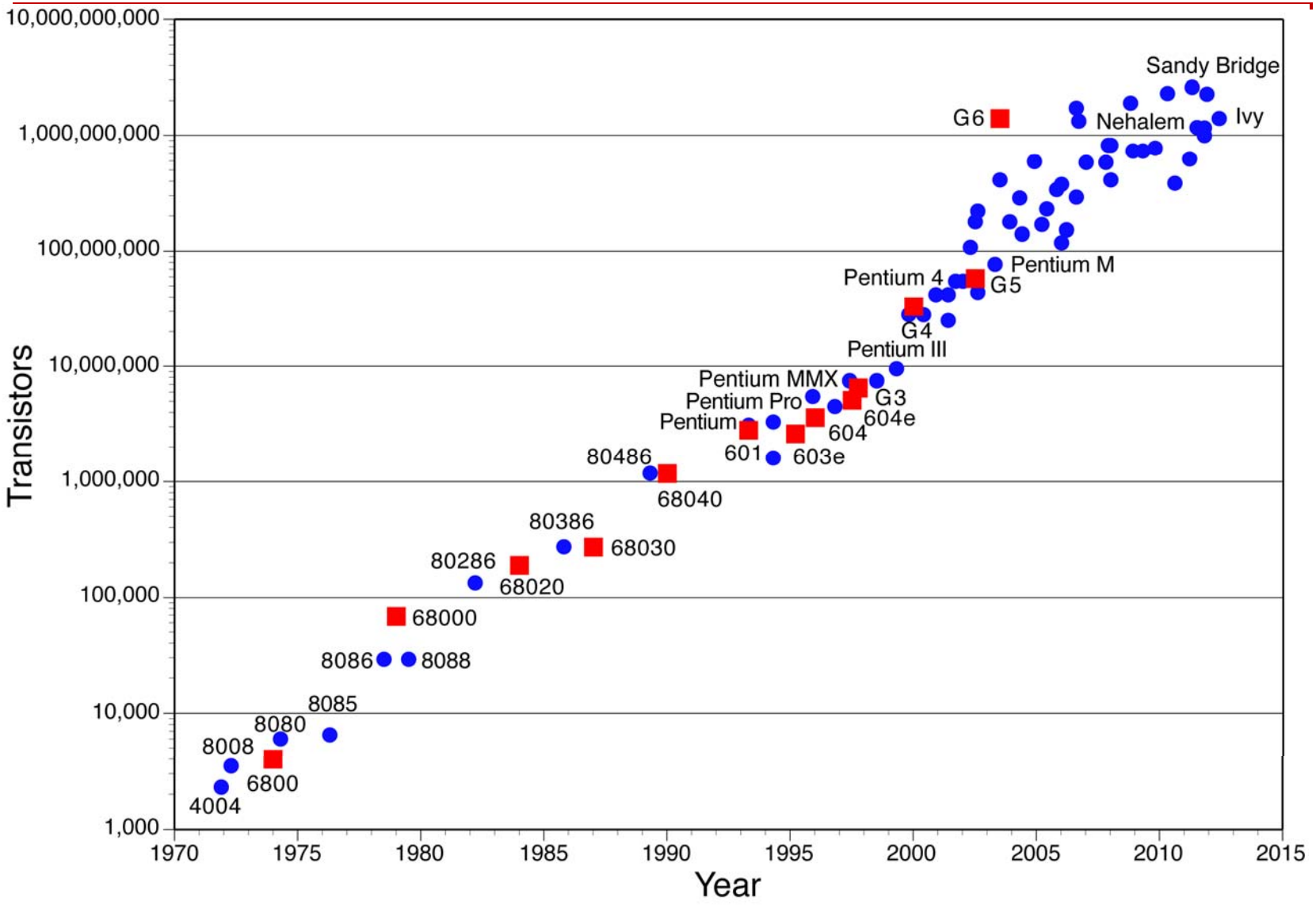


Design of Integrated Systems

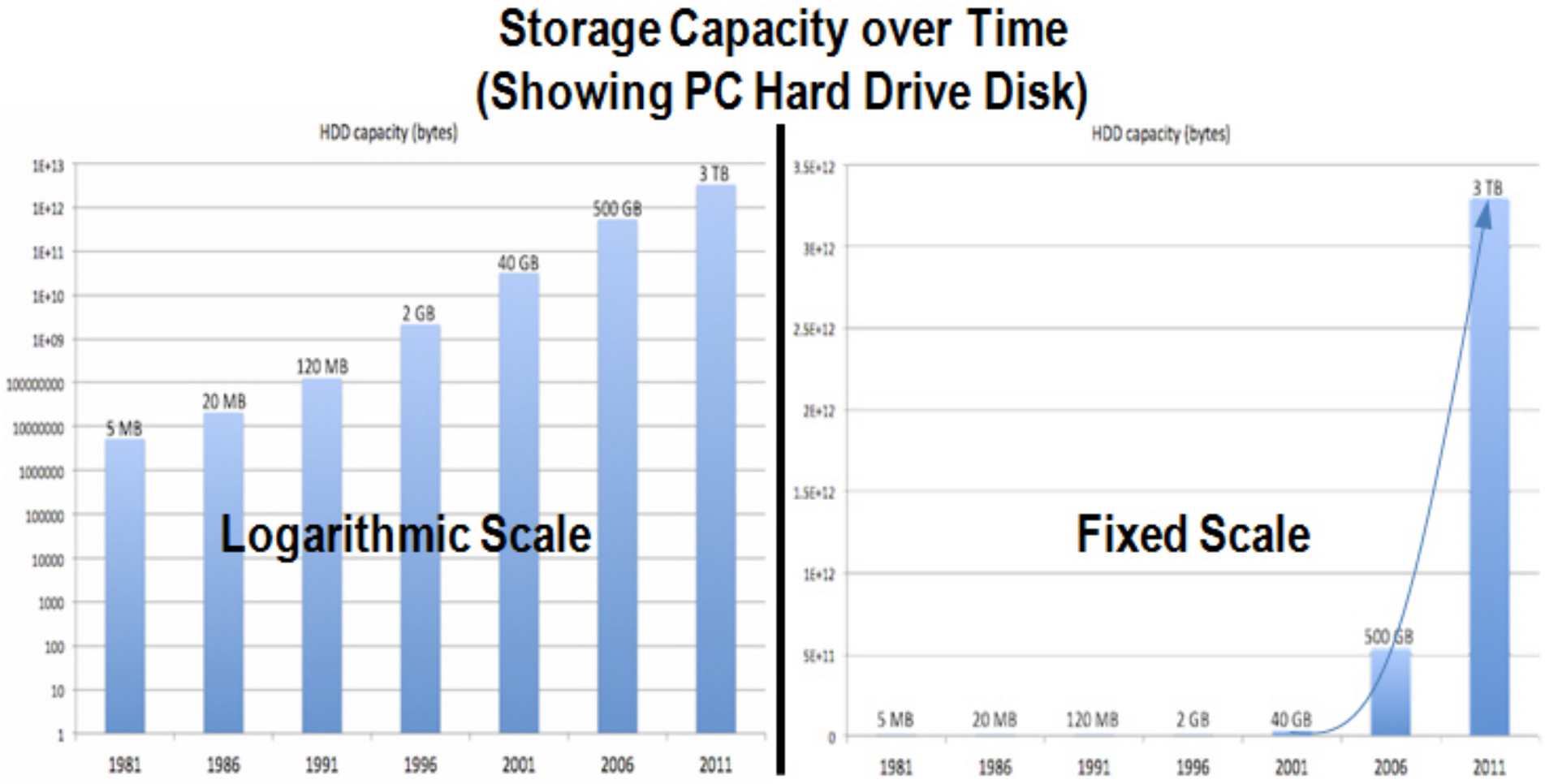
- ✓Design phases overlap to large degrees
- ✓Parallel changes on multiple levels, multiple teams
- ✓Tight scheduling constraints for product



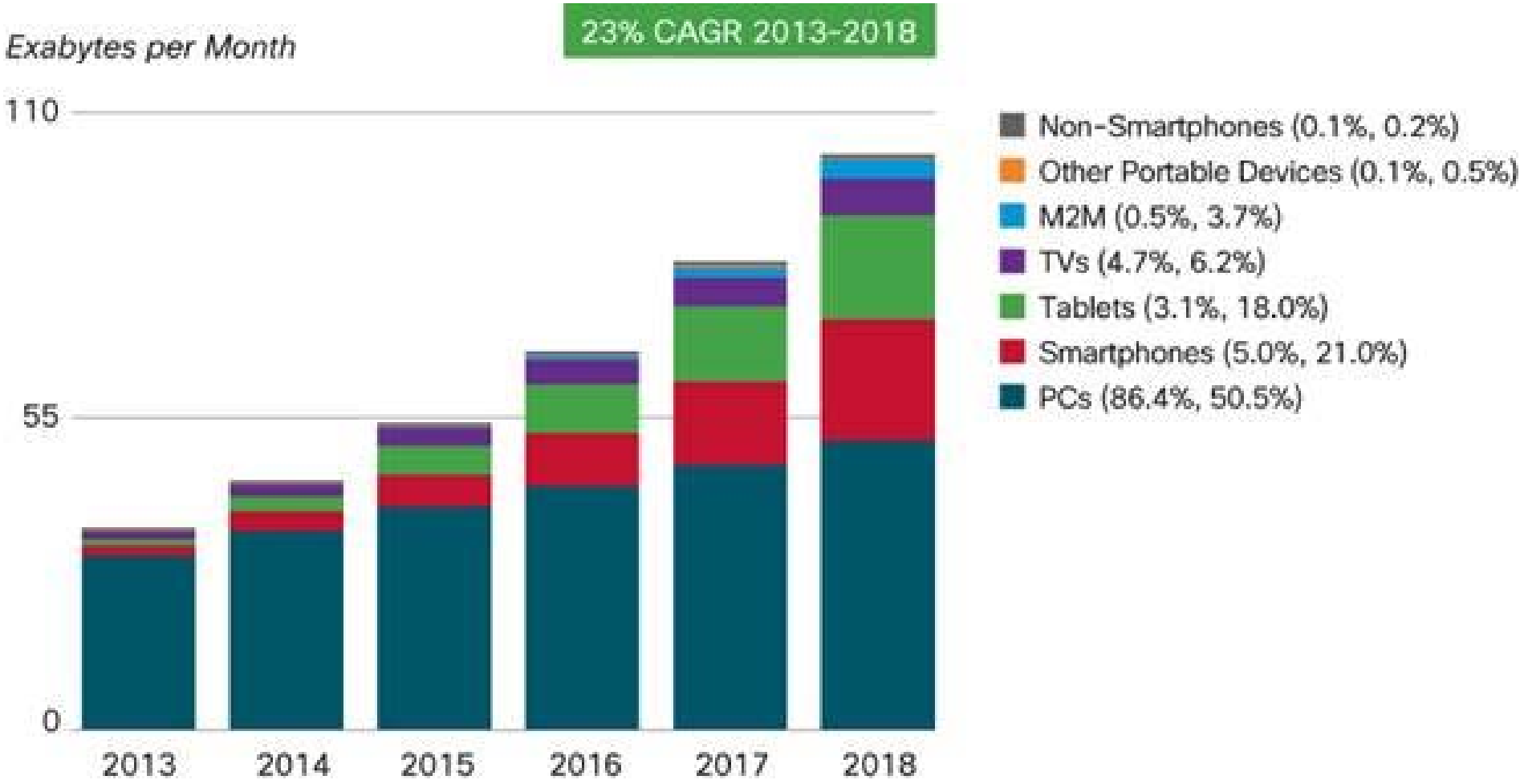
Processor Growth: Moore's Law



Storage Capacity



Global Internet Traffic by Device Type

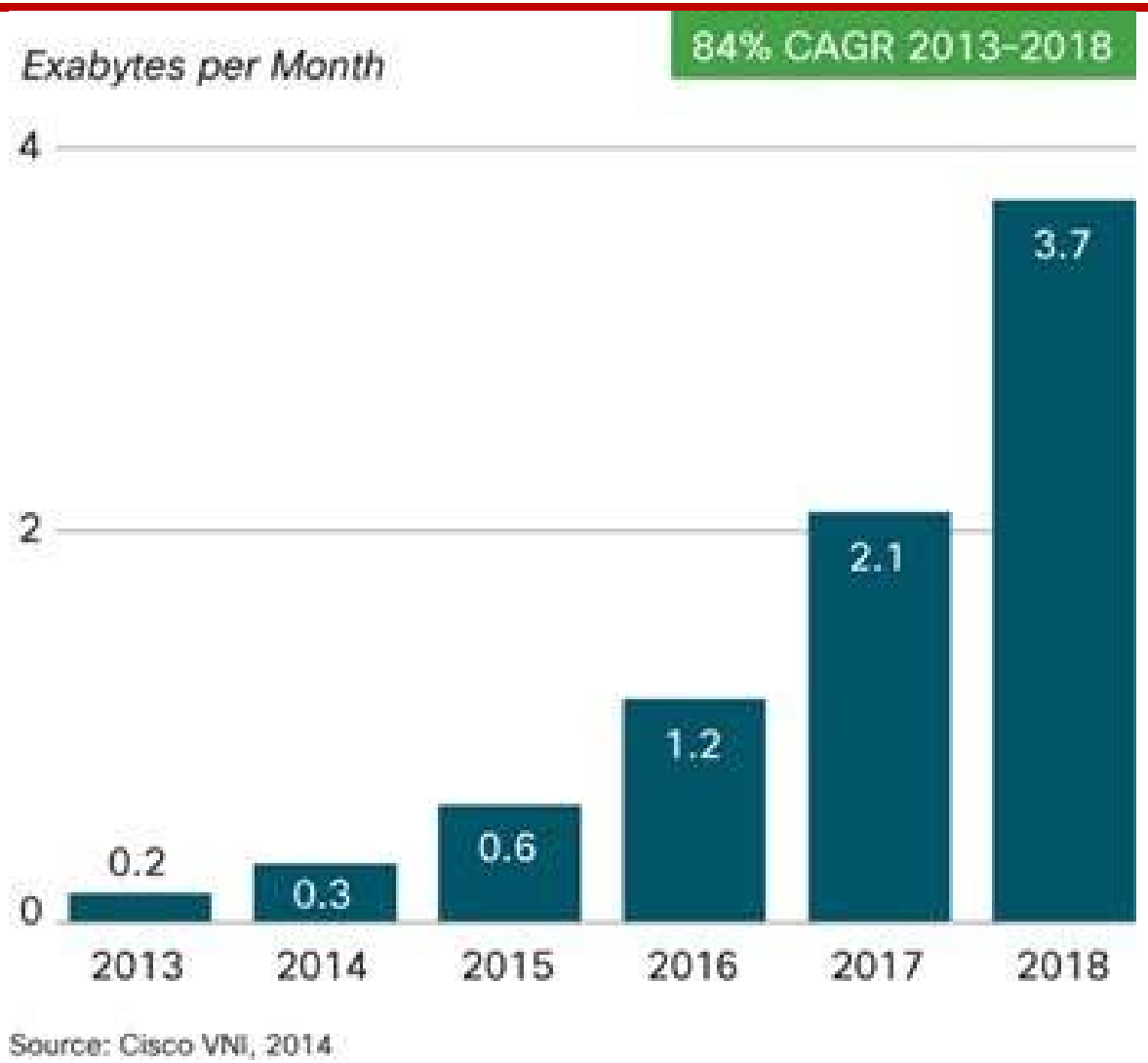


Source: Cisco VNI, 2014

The percentages in parentheses next to the legend denote the device traffic shares for the years 2013 and 2018, respectively.

Tablets are the fastest-growing device category with 29 percent CAGR (3.6-fold growth) over the forecast period, followed by machine-to-machine (M2M) connections with 26 percent CAGR (threefold growth).

M2M Traffic Growth



While the number of connections is growing threefold, global M2M IP traffic will grow 11-fold over this same period

Design Challenges

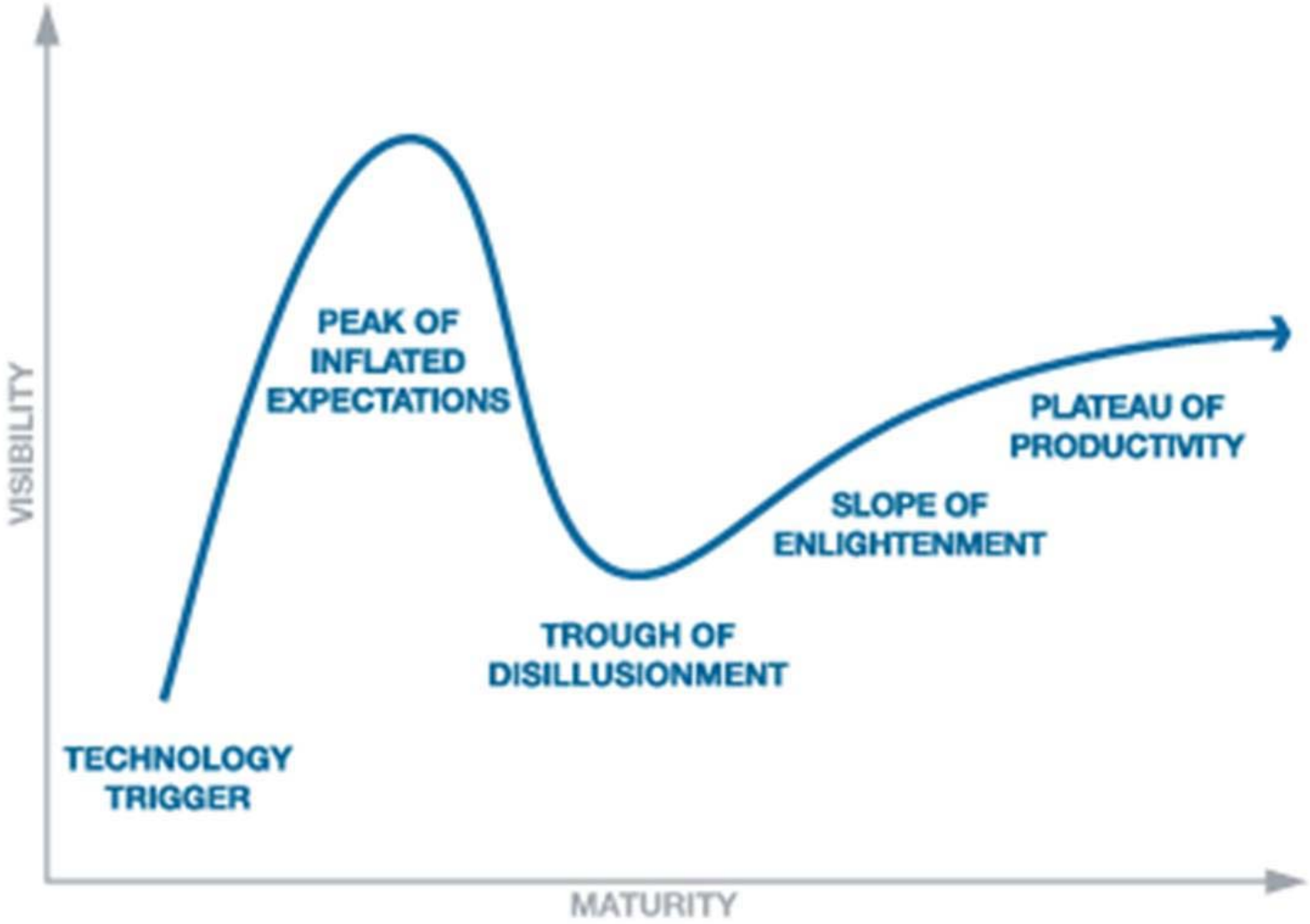
- ✓ Systems are becoming huge, design schedules are getting tighter
  - > 100 M gates becoming common for ASICs
  - > 0.4 M lines of C-code to describe system behavior
  - > 5 M lines of RLT code
- ✓ Design teams are getting very large for big projects
  - several hundred people
  - differences in skills
  - concurrent work on multiple levels
  - management of design complexity and communication very difficult
- ✓ Design tools are becoming more complex but still inadequate
  - typical designer has to run ~50 tools on each component
  - tools have lots of bugs, interfaces do not line up etc.

Design Challenges

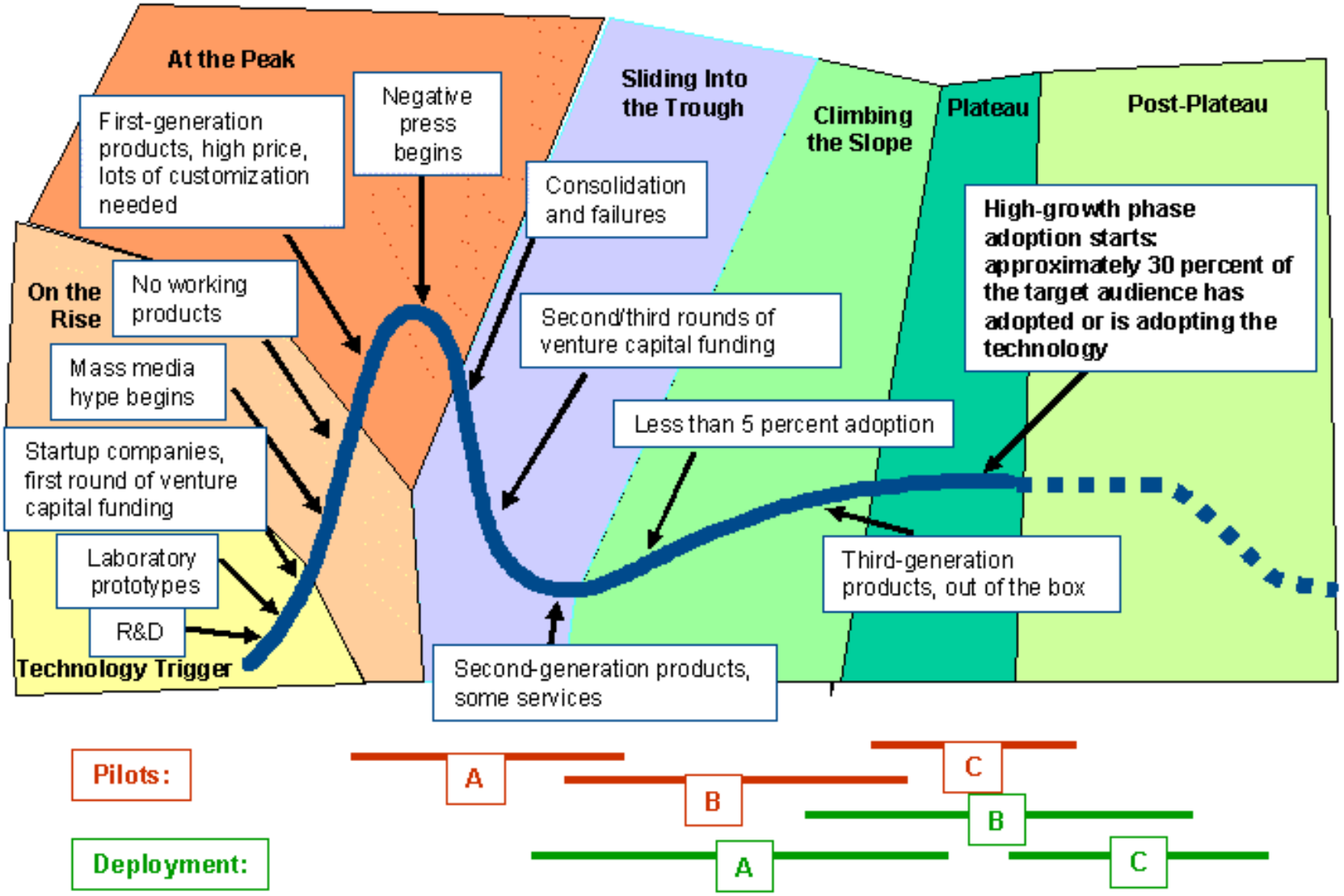
- ✓ Decision about design point very difficult
  - compromise between performance / costs / time-to-market
  - decision has to be made 2-3 years before design finished
  - design points are difficult to predict without actually doing the design
  - scheduling of product cycles

- ✓ Functional verification
  - simulation still main vehicle for functional verification but inadequate because of size of design space
  - results in bugs in released hardware that is very expensive to recover from (different in software)

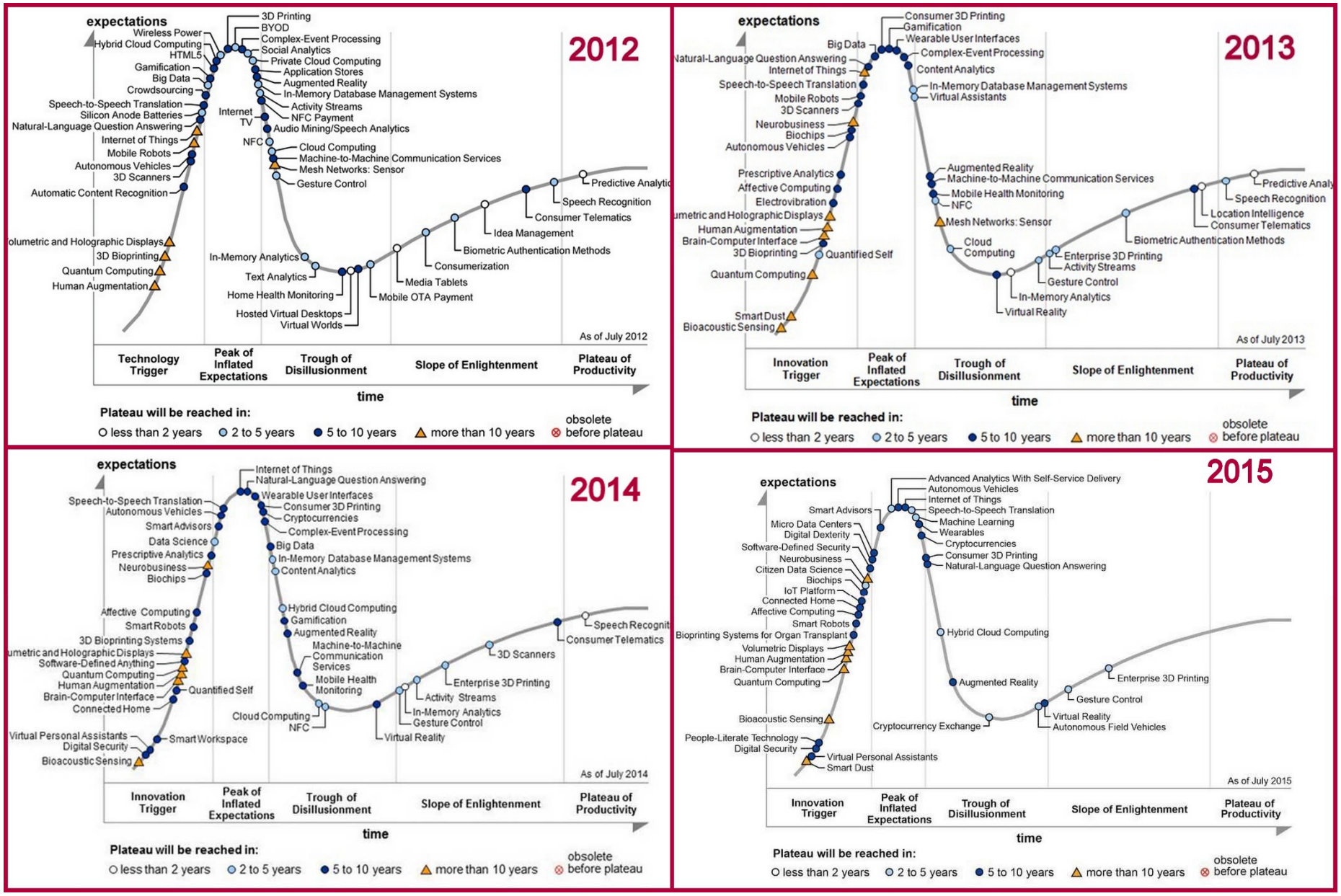
Gartner Hype Cycle



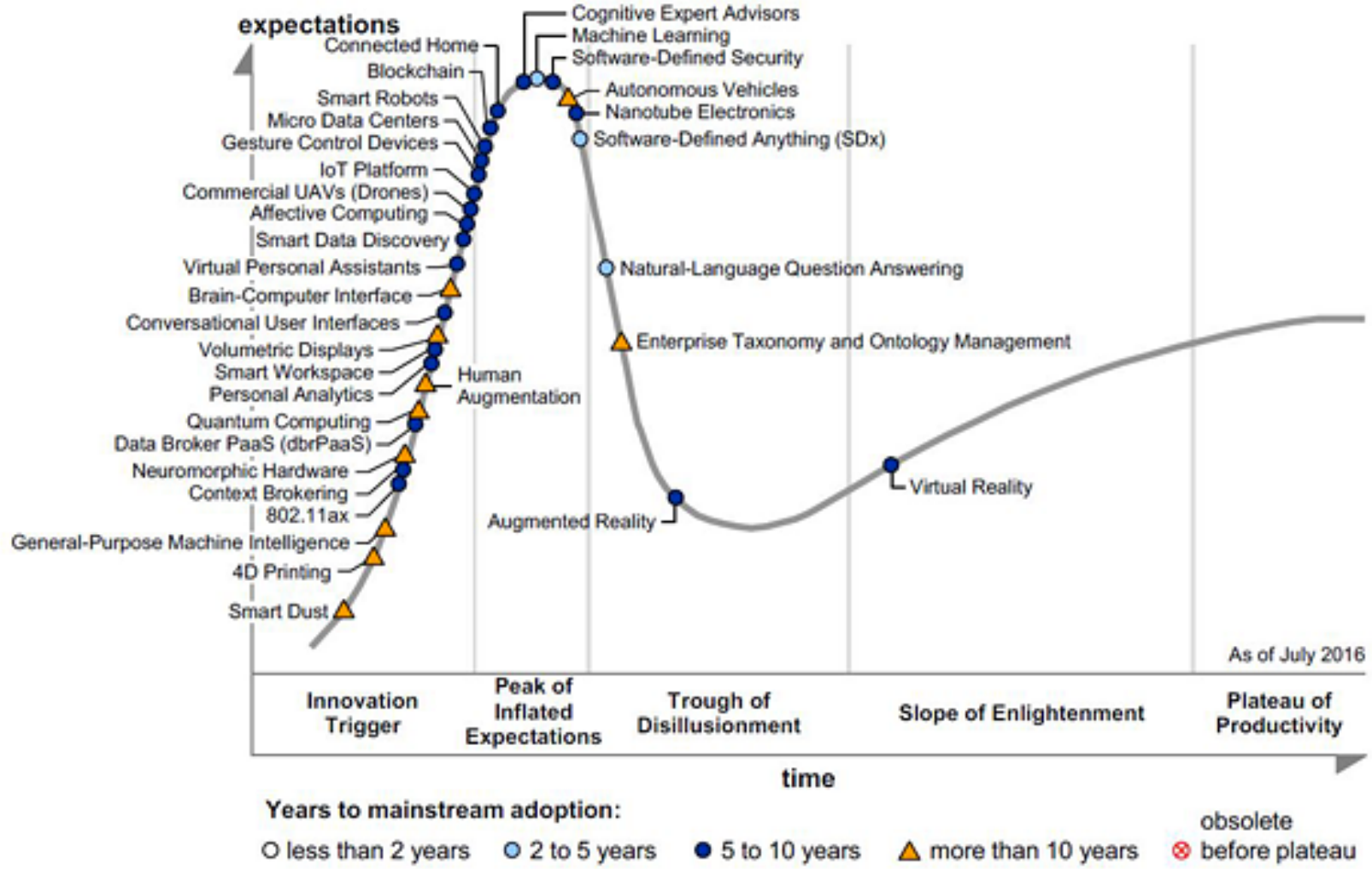
Gartner Hype Cycle



Gartner Hype Cycle 2012-15

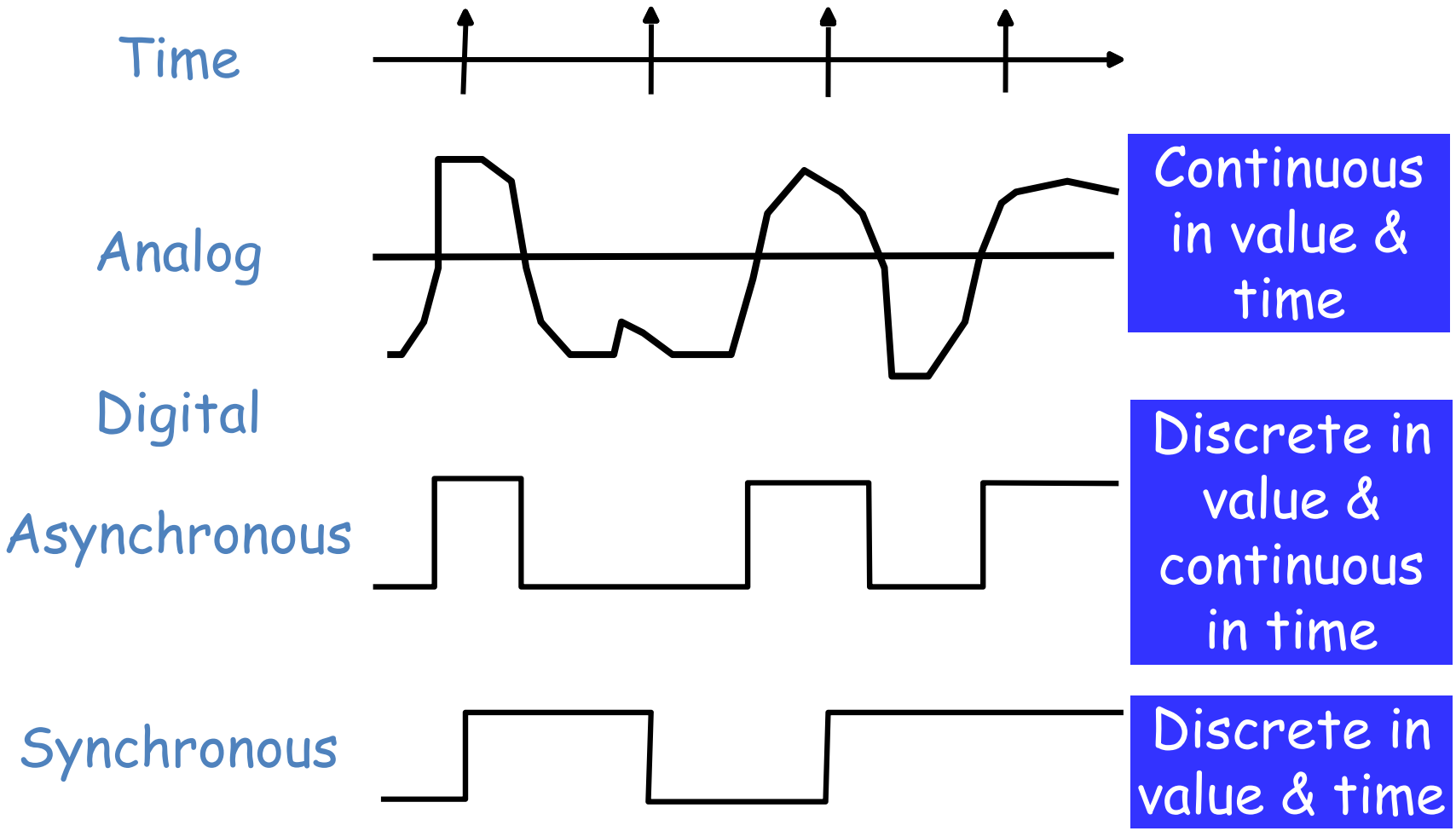


Gartner Hype Cycle 2016

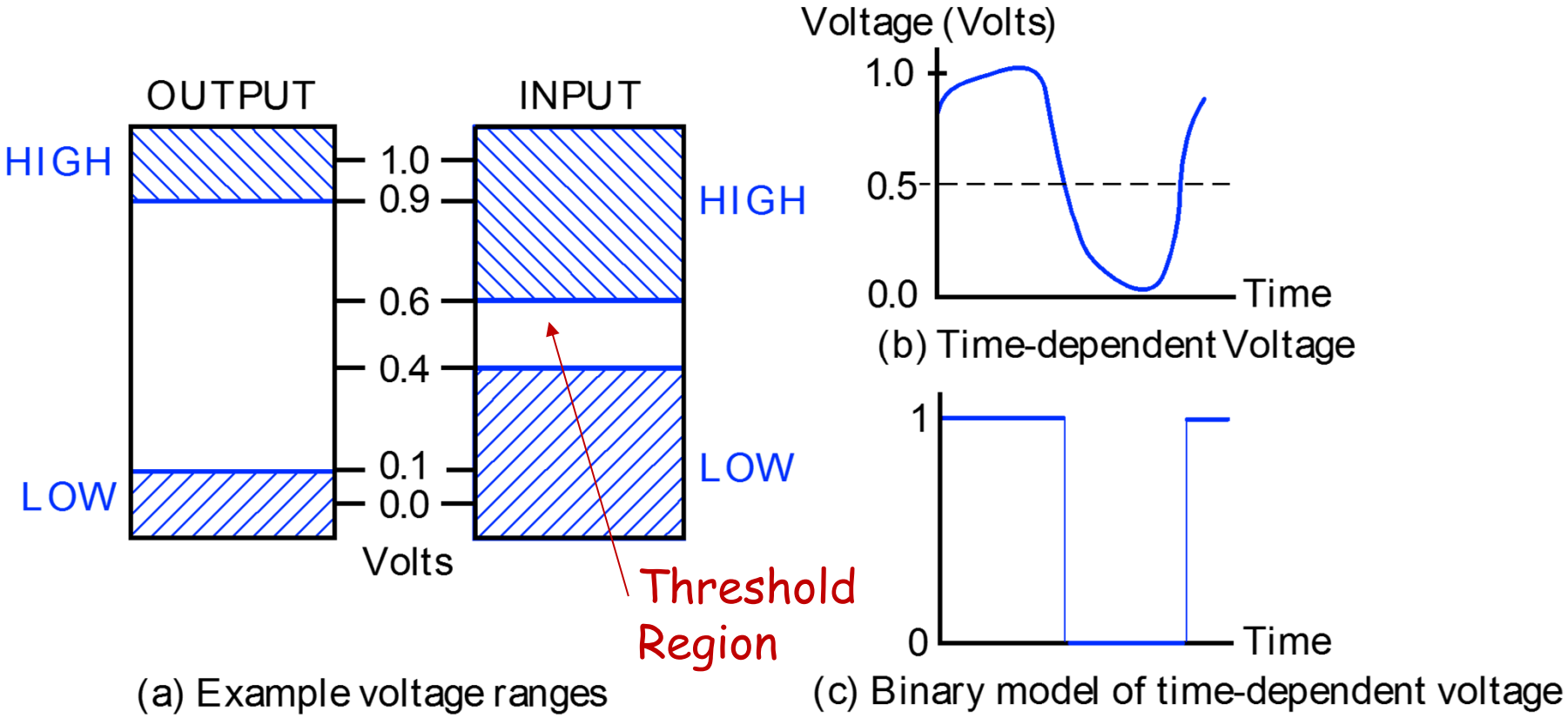


Source: Gartner (July 2016)

Signal Examples Over Time



Signal Example – Physical Quantity: Voltage



Binary Values: Other Physical Quantities

- ✓ What are other physical quantities represent 0 and 1?
  - CPU — Voltage
  - Disk — Magnetic Field Direction
  - CD — Surface Pits/Light
  - Dynamic RAM — Electrical Charge
  -
  -
  -