دوره مقدماتی آشنایی با FPGA و VHDL

مقدمه

محمدرضا عزیزی

امیرعلی ابراهیمی

بهار ۱۳۹۷

طراحی دیجیتال – مقدمه

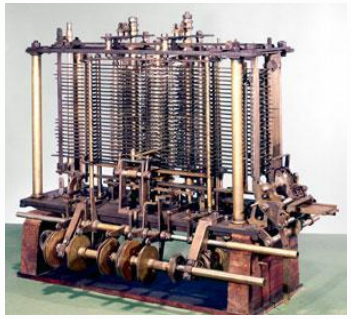
دیجیتال

- به انواعی از مدارهای الکترونیکی اشاره می‌کند که اطلاعات را فقط با استفاده از دو سطح ولتاژ ۰ و ۱ نمایش می‌دهند.

طراحی

- به پروسه اصولی ساختن مدارها به صورتی که نیازمندی های خاصی را جوابگو باشند و قیدهای خاصی را دارا باشند.
  - هزینه
  - کارایی
  - مصرف انرژی
  - وزن

طراحی دیجیتال – تاریخچه (۱)

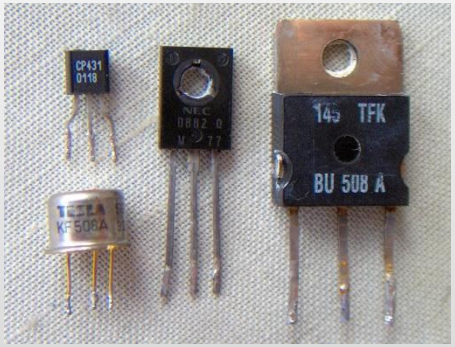


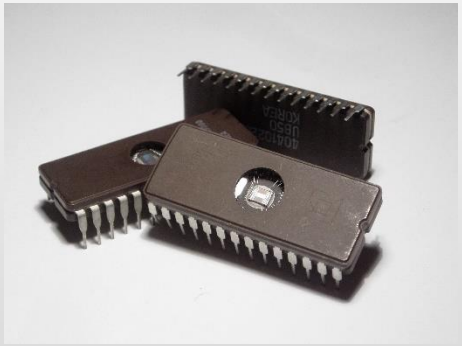
- **روند تکنولوژی در گذر زمان:**
  - ماشین های مکانیکی
  - ماشین های الکترومکانیکی
  - ماشین های الکترونیکی آنالوگ

- **مشکلات:**
  - دقت پایین
  - سرعت پایین
  - هزینه نگه‌داری بالا

طراحی دیجیتال – تاریخچه (۲)

- **اختراع لامپ خلاء و ترانزیستور**
  - انقلاب!
- **اختراع IC (Integrated Circuit):**
  - هر IC شامل چندین ترانزیستور
  - کاهش اندازه ترانزیستورها ← ساخت IC هایی با ملیارد ها ترانزیستور





نمایش سیستم (SYSTEM REPRESENTATION)

- یک سیستم دیجیتال **بزرگ**، **پیچیده** است. در پروسه توسعه و تولید یک سیستم دیجیتال، هر عمل به اطلاعات خاصی از سیستم نیاز دارد که این اطلاعات می‌تواند در بازه مشخصات کلی سیستم تا لی‌اوت اجزا سخت‌افزاری سیستم باشد.
- در نتیجه، سیستم به حالت‌های مختلفی توصیف می‌شود و مورد آزمون قرار می‌گیرد. به این روش‌ها، **نمایش سیستم** گوییم.
  - نمایش رفتاری (Behavioral view)
  - نمایش ساختاری (Structural view)
  - نمایش فیزیکی (Physical view)

نمایش سیستم (SYSTEM REPRESENTATION)

نمایش رفتاری (Behavioral view):

- توصیف سیستم از لحاظ کارکرد (function)
- در این حالت سیستم به عنوان یک **جعبه سیاه** در نظر گرفته می‌شود و به پیاده‌سازی داخلی آن پرداخته نمی‌شود.

نمایش ساختاری (Structural view):

- توصیف پیاده‌سازی داخلی سیستم
- بیان این که **چه** اجزائی وجود دارد و **چگونه** به هم متصل شده اند.

نمایش سیستم (SYSTEM REPRESENTATION)

➢ نمایش فیزیکی (Physical view):

- توصیف ویژگی‌های فیزیکی سیستم و افزودن اطلاعاتی به نمایش ساختاری
- مشخص کردن **اندازه فیزیکی** اجزا، **مکان فیزیکی** اجزا بر روی بورد یا ویفر سیلیکونی، **مسیر** (path) **فیزیکی** موجود بین اجزایی که به یکدیگر متصل شده اند.

انتزاع (ABSTRACTION)

- مشخص کردن **میزان پیچیدگی** و **وجه**های مورد نیاز از یک سیستم
- انتزاع، **مدل ساده‌شده** سیستم است که تنها اطلاعات و ویژگی‌هایی که انتخاب شده است را نشان می‌دهد و بقیه اطلاعات را نادیده می‌گیرد.
- انتزاع ← ساده شدن آنالیز و پروسه طراحی ← امکان طراحی سیستم های پیچیده تر

➢ **یک مثال:**

- یک چیپ شامل میلیون ها ترانزیستور است. پردازش کردن این حجم از اطلاعات به طور مستقیم توسط یک انسان و یا حتی یک کامپیوتر، غیر ممکن است.
- **راه حل: توصیف سیستم در سطح‌های مختلفی از انتزاع**

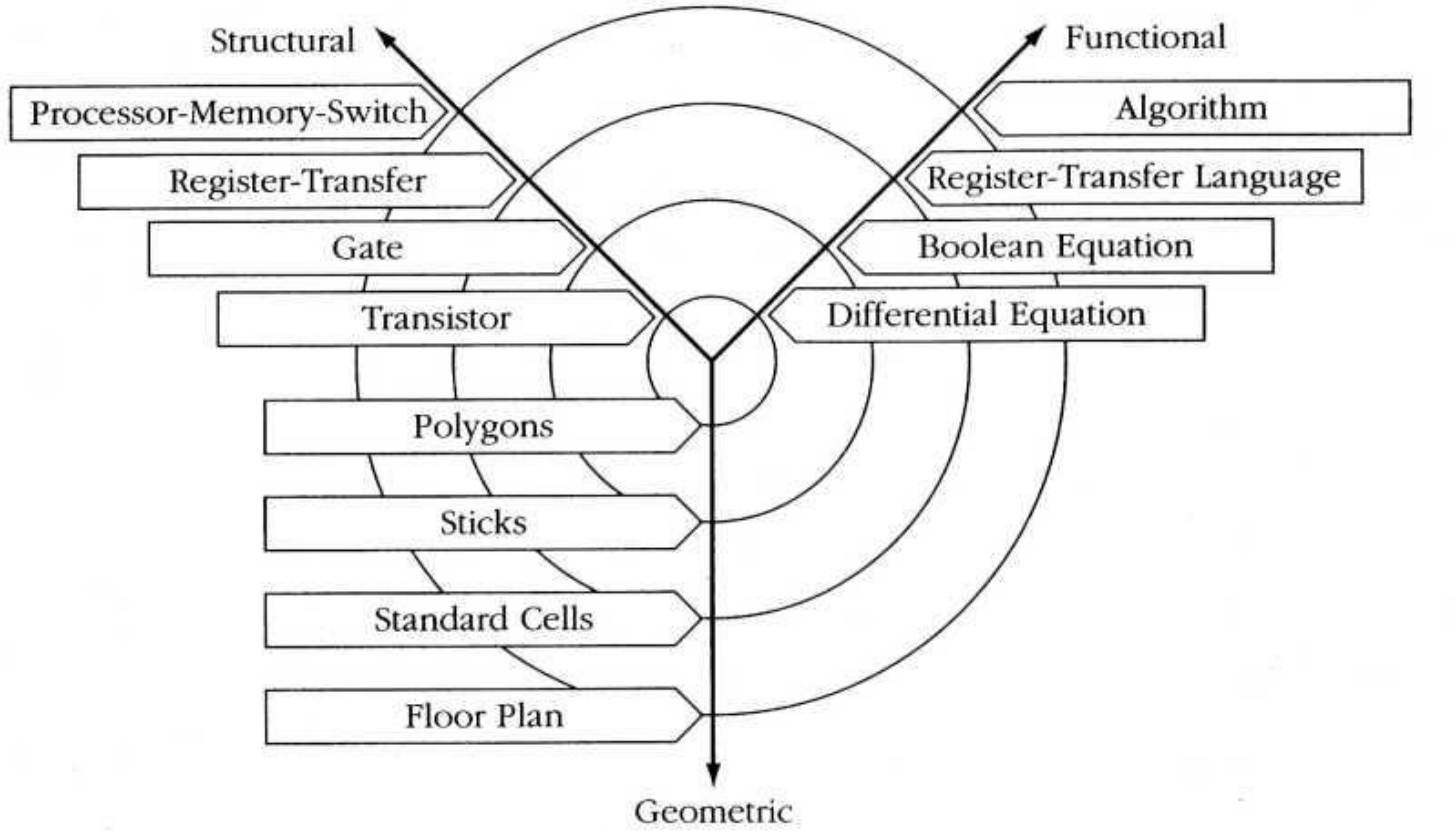
➢ **معمولا فرایند طراحی، از یک انتزاع سطح بالا شروع می‌شود.**

سطوح انتزاع (LEVELS OF ABSTRACTION)

Table 1.2 Characteristics of each abstraction level

	Typical blocks	Signal representation	Time representation	Behavioral description	Physical description
Transistor	transistor, resistor	voltage	continuous function	differential equation	transistor layout
Gate	and, or, xor, flip-flop	logic 0 or 1	propagation delay	Boolean equation	cell layout
RT	adder, mux, register	integer, system state	clock tick	extended FSM	RT-level floor plan
Processor	processor, memory	abstract data type	event sequence	algorithm in C	IP-level floor plan

سطوح انتزاع (LEVELS OF ABSTRACTION)



Domains and levels of abstraction. The radial axes show the three different domains of modeling. The concentric rings show the levels of abstraction, with the more abstract levels on the outside and more detailed levels toward the center.

سیستم نهفته (EMBEDDED SYSTEM)

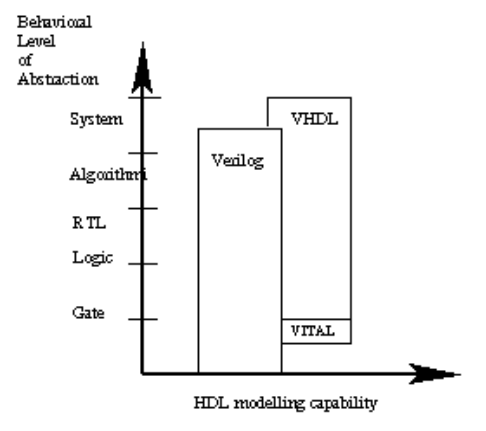
تعریف:

- رایانه‌هایی هستند که برای کنترل یک سیستم بزرگ و مشخص طراحی شده‌اند.
- سامانه نهفته معمولا به عنوان قسمتی از یک سامانه بزرگ که اغلب دارای سخت‌افزارها و قسمت‌های مکانیک مختلفی است جاسازی می‌شود.
- برخلاف یک رایانه چند منظوره، مانند یک کامپیوتر شخصی که به شکلی انعطاف پذیر (از نظر معماری پردازنده) طراحی شده است که قسمت بزرگی از نیازهای مصرف کننده‌ها را برآورده کند برای انجام کار مشخصی طراحی شده اند.
- مشخصه کلیدی این سامانه‌ها، طراحی اختصاصی برای انجام یک کار مشخص است.

زبان‌های توصیف سخت‌افزار

- VHDL vs. Verilog:

Verilog	VHDL
توسعه داده شده با اهداف تجاری	توسعه داده شده توسط وزارت دفاع آمریکا
ساختار بر اساس زبان C	ساختار بر اساس زبان Ada
بیشتر در اروپا	در سطح جهان
Case Sensitive است	Case Sensitive نیست



زبان‌های توصیف سخت‌افزار

- نمونه کد VHDL:
  - VHISC(Very High Speed Integrated Circuit) HDL

```
reg1: process (rst, clk)
    begin
        if  rst = '1' then
            q_reg <= (others => '0');
            q_i <= (others => '0');
        elsif  rising_edge(clk) then
            if  s_l = '1' then
                q_i(0) <= q_i(7);
            loop1:  for i in 6 downto 0 loop
                        q_i(i + 1) <= q_i(i);
                end loop loop1;
                q_reg <= y;
            else
                q_i <= q_reg;
                q_reg <= y;
            end if;
        end if;
    end process reg1;
```

زبان‌های توصیف سخت‌افزار

- نمونه کد Verilog:

```
always @(posedge CLK or posedge RST)
begin
          if (RST) begin
                  q_reg = 0;
                  Q = 0;
          end else if (S_L) begin
                  Q[7:0] = {Q[6:0],Q[7]};
                  q_reg = Y;
          end else begin
                  Q = q_reg;
                  q_reg = Y;
          end
end
```

زبان‌های توصیف سخت‌افزار

- SystemVerilog:
  - شی گرا

```
property p_push_error;
    @ (posedge clk)
       not (b_if.push && b_if.full && !b_if.pop);
  endproperty : p_push_error
  ap_push_error_1 : assert property (p_push_error);

  property p_pop_error;
   @ (posedge clk)
       not (b_if.pop && b_if.empty);
  endproperty : p_pop_error
  ap_pop_error_1 : assert property (p_pop_error);

always_ff @ (posedge clk) begin
    b_if.error <= (b_if.pop && b_if.empty) || (b_if.push && b_if.full && !b_if.pop
```

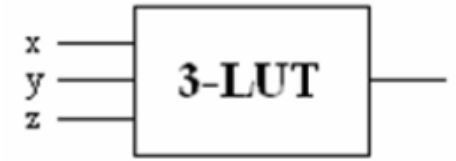
FPGA

Field-Programmable Gate Array (آرایه گیت های قابل برنامه ریزی)

- . یک مدار مجتمع است که می توان آن را پس از اتمام فرآیند تولید ، مطابق نیاز طراح برنامه ریزی نمود و روابط منطقی بین پایه های ورودی و خروجی را تغییر داد.
- برنامه نویسی با استفاده از زبان های توصیف سخت افزار
- تفاوت با میکرو کنترلر: میکرو کنترلر دارای پردازنده است و دستورات را به صورت sequential با استفاده از پردازنده انجام می‌دهد در حالی که FPGA تراشه ای بدون پردازنده است!

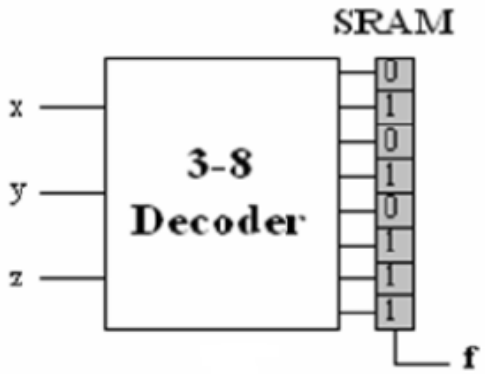
- نحوه عملکرد FPGA:

$$f=xy+z$$



جدول درستی تابع f

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



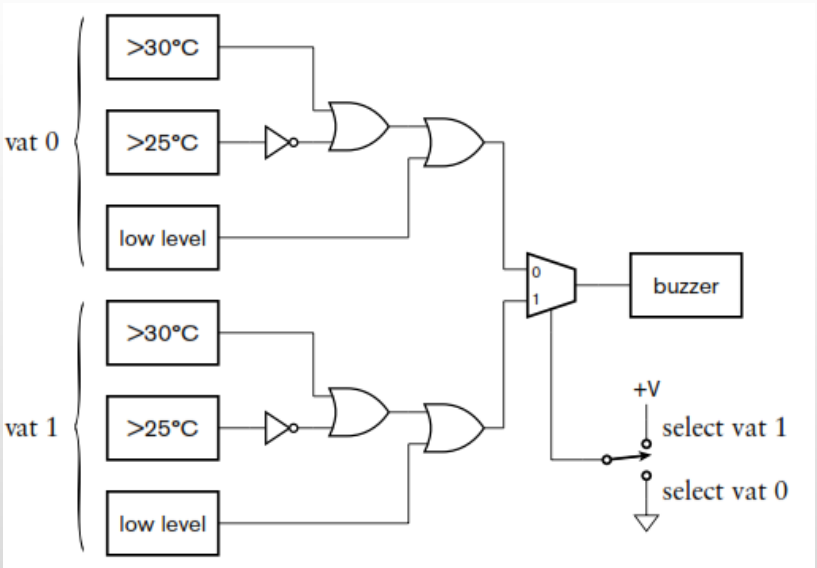
FPGA

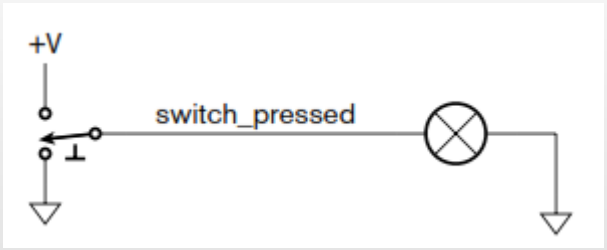
Common FPGA Applications:

- Aerospace and Defense
  - Avionics/DO-254
  - Communications
  - Missiles & Munitions
  - Secure Solutions
  - Space
- Medical Electronics
- ASIC Prototyping
- Audio
  - Connectivity Solutions
  - Portable Electronics
  - Radio
  - Digital Signal Processing (DSP)
- Automotive
  - High Resolution Video
  - Image Processing
  - Vehicle Networking and Connectivity
  - Automotive Infotainment
- Broadcast
  - Real-Time Video Engine
  - EdgeQAM
  - Encoders
  - Displays
  - Switches and Routers
- Consumer Electronics
  - Digital Displays
  - Digital Cameras
  - Multi-function Printers
  - Portable Electronics
  - Set-top Boxes
- Data Center
  - Servers
  - Security
  - Routers
  - Switches
  - Gateways
  - Load Balancing
- High Performance Computing
  - Servers
  - Super Computers
  - SIGINT Systems
  - High-end RADARs
  - High-end Beam Forming Systems
  - Data Mining Systems
- Industrial
  - Industrial Imaging
  - Industrial Networking
  - Motor Control
- Medical
  - Ultrasound
  - CT Scanner
  - MRI
  - X-ray
  - PET
  - Surgical Systems
- Scientific Instruments
  - Lock-in amplifiers
  - Boxcar averagers
  - Phase-locked loops
- Security
  - Industrial Imaging
  - Secure Solutions
  - Image Processing
- Video & Image Processing
  - High Resolution Video
  - Video Over IP Gateway
  - Digital Displays
  - Industrial Imaging
- Wired Communications
  - Optical Transport Networks
  - Network Processing
  - Connectivity Interfaces
- Wireless Communications
  - Baseband
  - Connectivity Interfaces
  - Mobile Backhaul
  - Radio

یادآوری: مدارهای ترکیبی (COMBINATIONAL)

- **مقدار خروجی مدار در هر زمان از ترکیب ورودی‌های مدار در آن لحظه به دست می‌آید.**
  - زمان و مقدار خروجی در لحظات قبل تاثیری بر خروجی مدار ندارد.

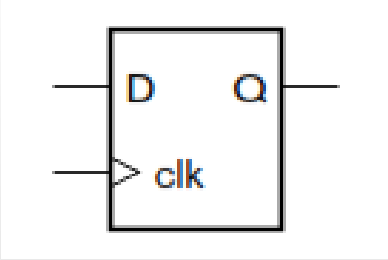




یادآوری: مدارهای ترتیبی (SEQUENTIAL)

- مقدار خروجی مدار در هر زمان از ترکیب ورودی‌های مدار در آن لحظه و وضعیت کنونی (که وضعیت کنونی نیز تابع ورودی های پیشین است) به دست می‌آید.

- مثال: فیلپ‌فلاپ



پایان مقدمه