آزمایشگاه معماری کامپیوتر

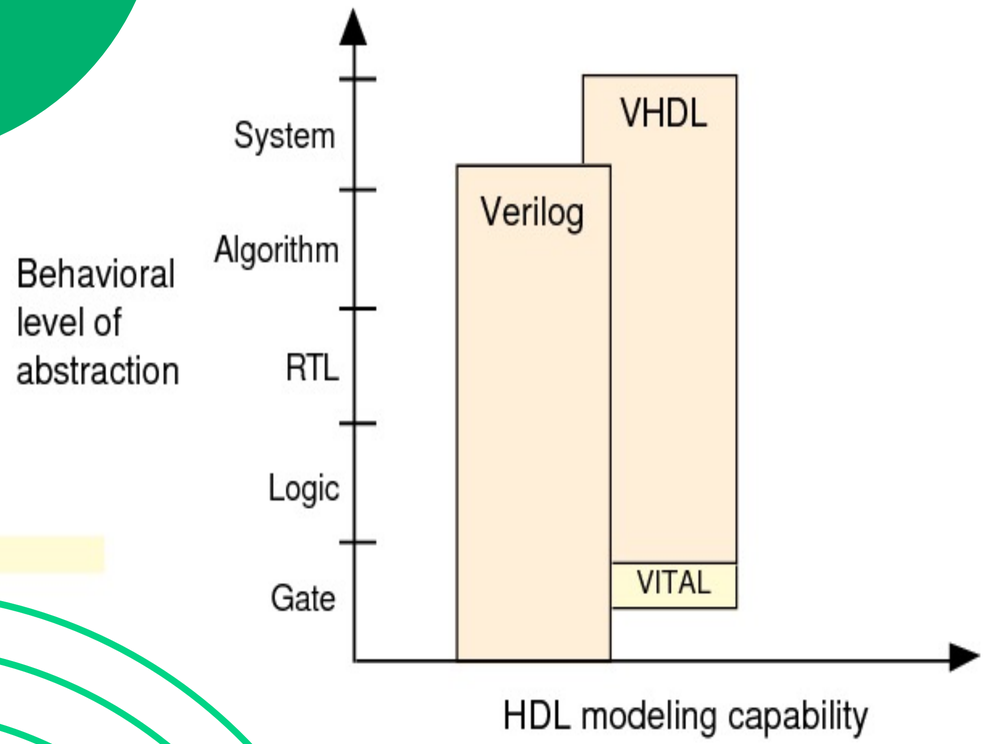
جلسه اول

- پیاده‌سازی گیت‌های and, or , xor
- پیاده‌سازی نیم‌جمع کننده با استفاده از گیت
- پیاده‌سازی یک تمام جمع کننده با استفاده از نیم‌جمع کننده و گیت or

VHDL چیست؟

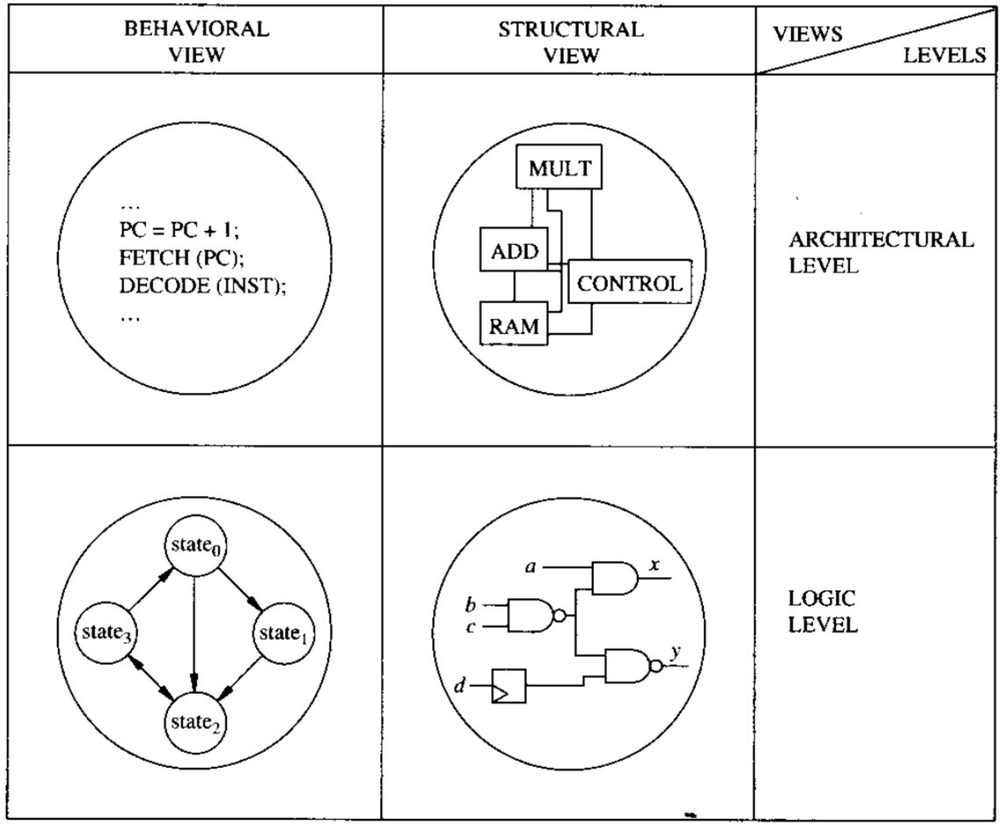
VHDL مخفف **V**ery High Speed Integrated Circuits **H**ardware **D**escription **L**anguage (**V**HSIC)، در دهه‌ی ۸۰ میلادی در وزارت دفاع آمریکا و با همکاری IEEE به منظور توسعه مدارهای مجتمع با سرعت بالا، طراحی شد.

امروزه از این زبان به عنوان یک زبان دارای استاندارد صنعتی برای توصیف سیستم‌های دیجیتال استفاده می‌شود.

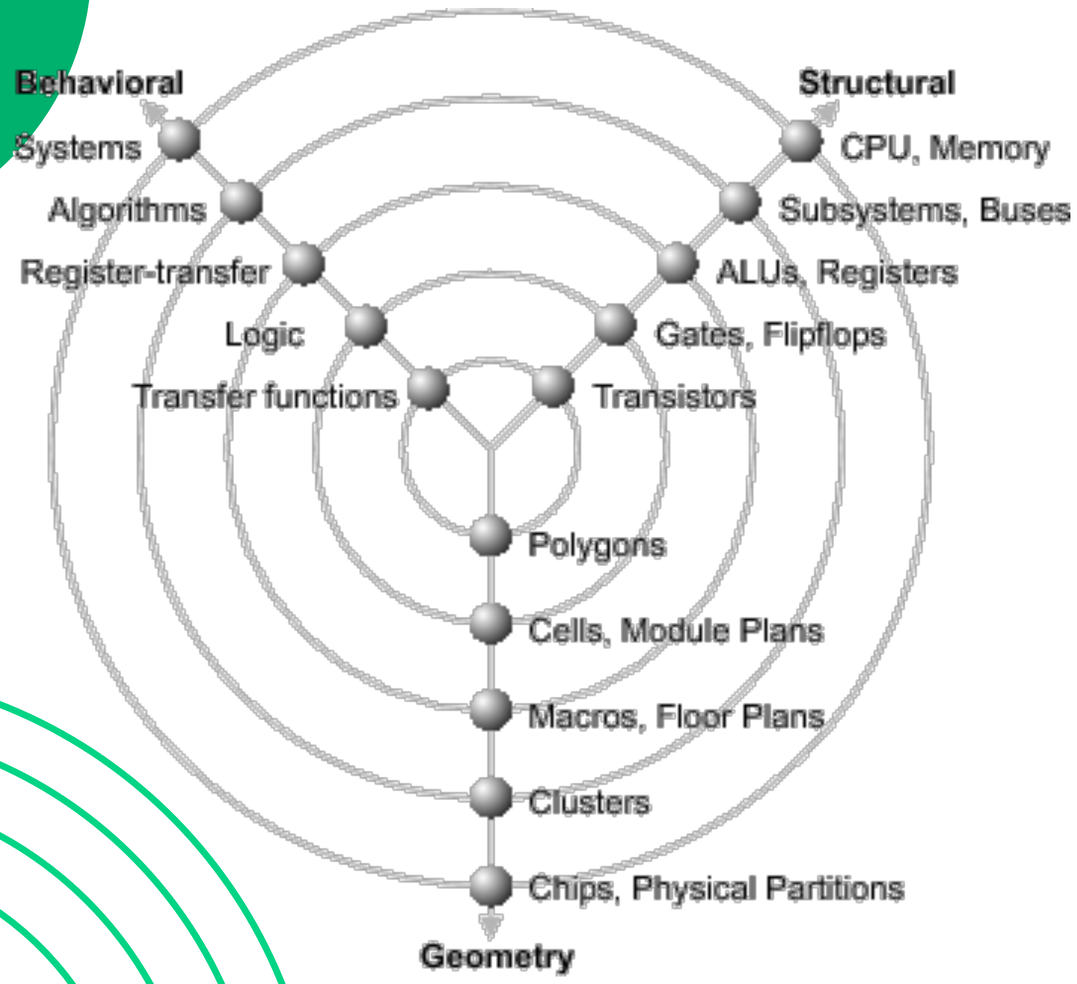


سطوح انتزاع

- استفاده از سطوح انتزاع باعث می‌شود تا توصیف سیستم‌های پیچیده قابل انجام شود.
- بالاترین سطح انتزاع، سطح رفتاری (Behavioral) است که سیستم را درقالب این‌که چگونه رفتار می‌کند توصیف می‌کند و به اجزا و ارتباطات میان آن‌ها اهمیتی نمی‌دهد.
- سطح ساختاری (Structural)، سیستم را در قالب مجموعه‌ای از اجزا، گیت‌ها و ارتباطات میانشان تعریف می‌کند، توصیف ساختاری را با توصیف شماتیک و ارتباطات میان گیت‌ها تمایسه می‌کنند.

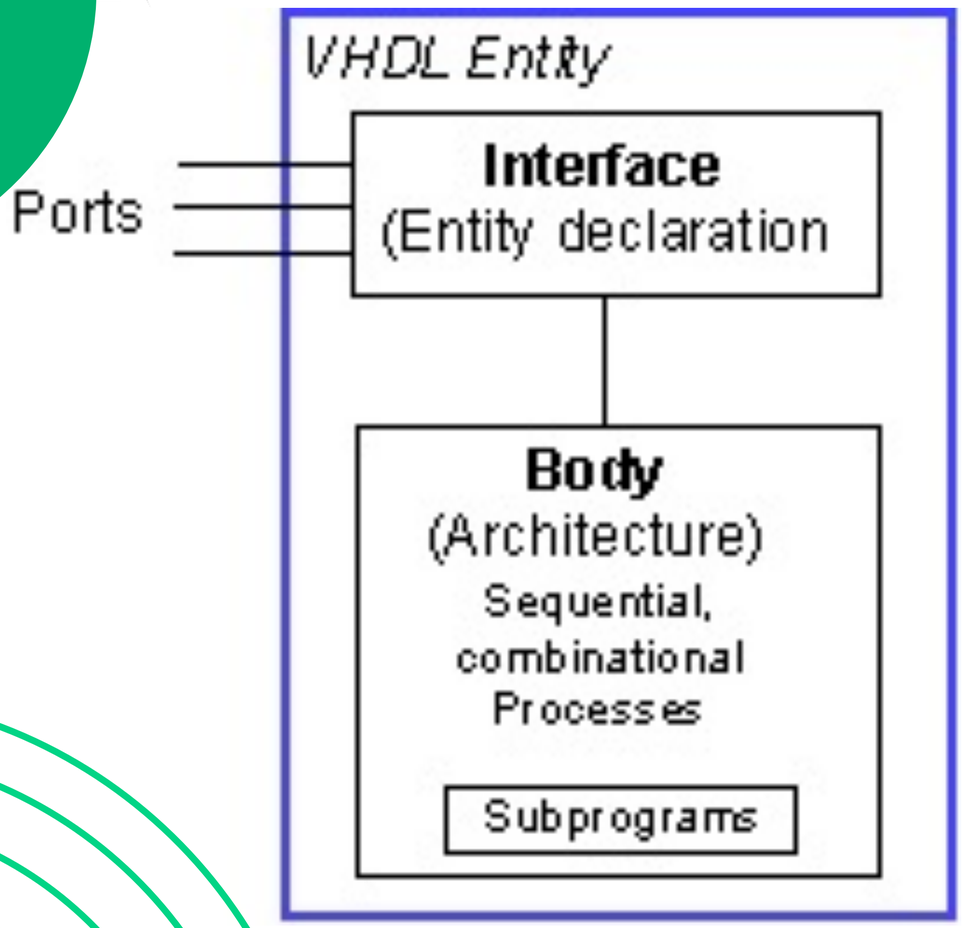


نمودار Gajski–Kuhn



- VHDL این امکان را فراهم می‌سازد تا سیستم دیجیتال در سطح ساختار یا رفتاری تعریف شود.

توصیف VHDL



- ساختار یک سیستم دیجیتال در VHDL

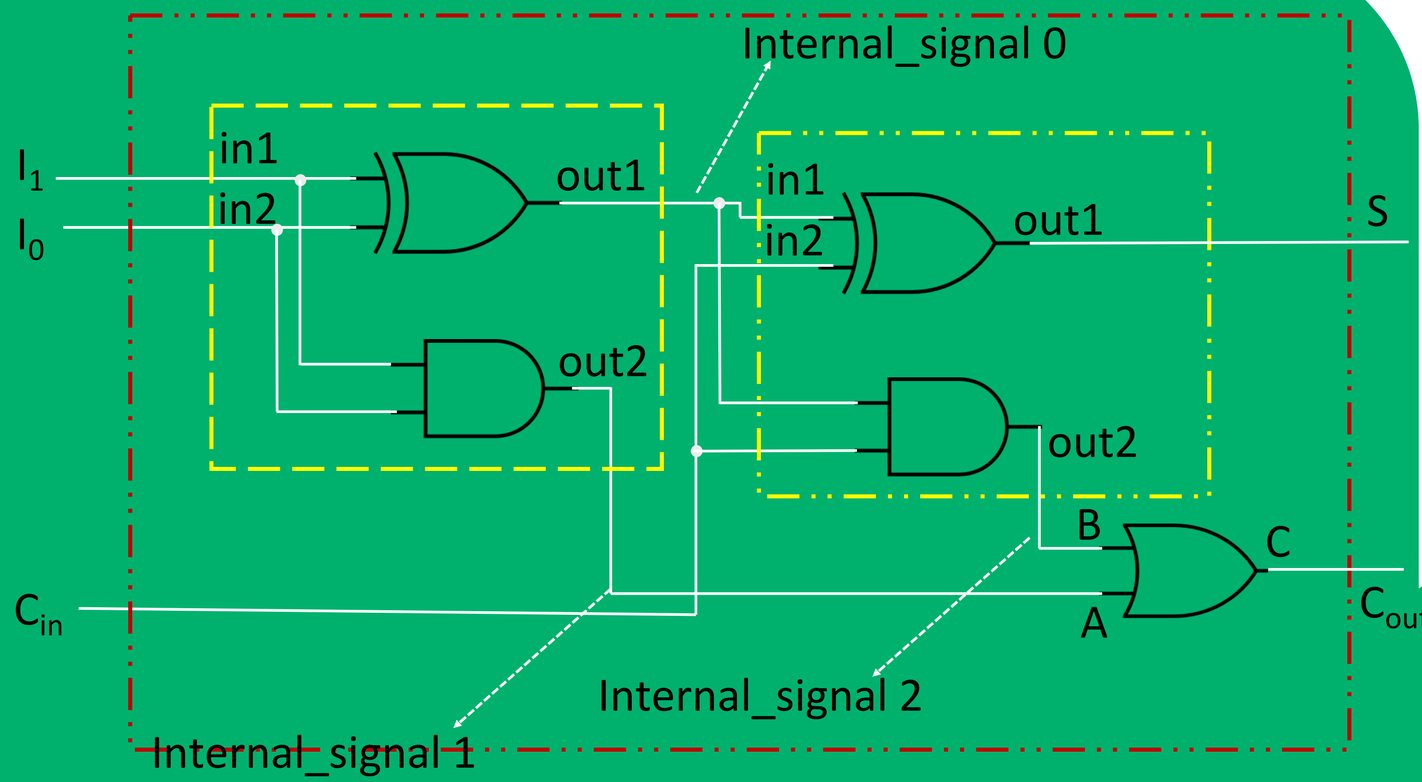
قواعد کلی VHDL

- VHDL دارای Keywordهایی است که نمی‌توان آن‌ها را به عنوان نام سیگنال و شناسه تعریف کرد.
- کلمات کلیدی و شناسه‌های تعریف شده توسط کاربر Case Insensitive هستند.
- استفاده از (--) باعث می‌شود تا نوشته‌های بعد از آن توسط کامپایلر نادیده گرفته شوند.
- VHDL همیشه روی نوع اشیاء حساس است و نیاز دارد که نوع تمامی اشیاء توسط کاربر تعریف شود.

ساختار VHDL

```
entity NAME_OF_ENTITY is [ generic generic_declarations);
    port (signal_names: mode type;
          signal_names: mode type;
              :
          signal_names: mode type);
end [NAME_OF_ENTITY] ;
```

آزمایش اول

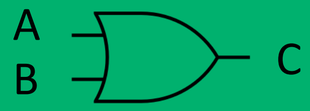

Internal_signal 0
I_1
I_0
in1
in2
out1
out2
in1
in2
out1
out2
S
B
C
A
C_{out}
C_{in}
Internal_signal 2
Internal_signal 1
Full adder
half adder

آزمایش اول

```
Library IEEE;
USE IEEE.std_logic_1164;
Entity and_gate is
Port (
            A, B: in std_logic;
            C    : out std_logic
);
End Entity and_gate;

Architecture gatelevel of and_gate is
Begin
C <= A and B;
End gatelevel;
```

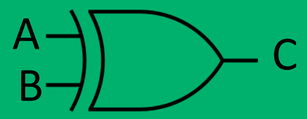
آزمایش اول

```
Library IEEE;
USE IEEE.std_logic_1164;
Entity or_gate is
Port (
            A, B: in std_logic;
            C    : out std_logic
);
End Entity or_gate;

Architecture gatelevel of or_gate is
Begin
C <= A or B;
End gatelevel;
```

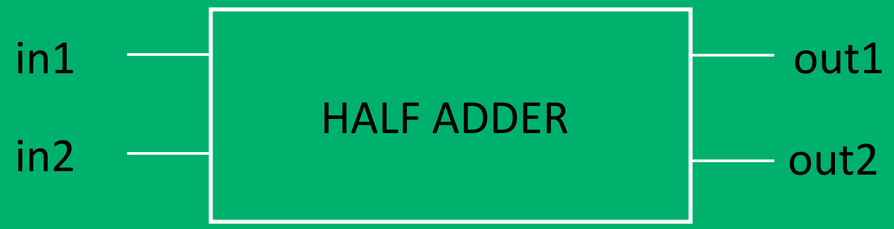


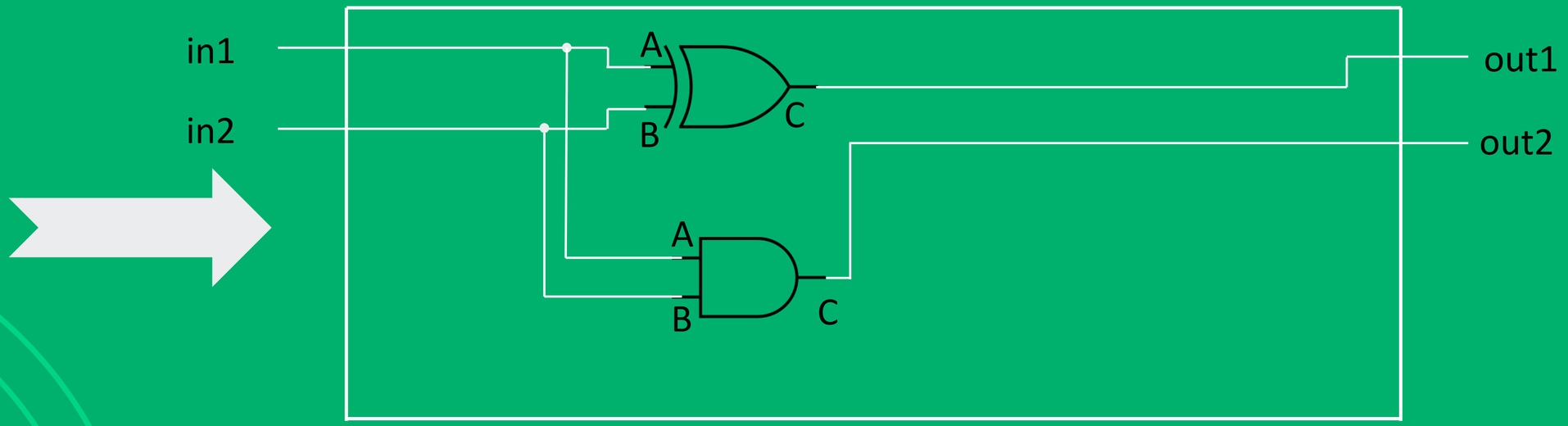
آزمایش اول

```
Library IEEE;
USE IEEE.std_logic_1164;
Entity xor_gate is
Port (
            A, B: in std_logic;
            C    : out std_logic
);
End Entity xor_gate;

Architecture gatelevel of xor_gate is
Begin
C <= A xor B;
End gatelevel;
```



آزمایش اول





آزمایش اول

```
Library IEEE;
USE IEEE.std_logic_1164;
Entity and_gate is
Port (
        A, B: in std_logic;
        C    : out std_logic
);
End Entity and_gate;

Architecture gatelevel of and_gate
is
Begin
C <= A and B;
End gatelevel;
```

```
Library IEEE;
USE IEEE.std_logic_1164;
Entity xor_gate is
Port (
        A, B: in std_logic;
        C    : out std_logic
);
End Entity xor_gate;

Architecture gatelevel of xor_gate is
Begin
C <= A xor B;
End gatelevel;
```

آزمایش اول

```
Library IEEE;
USE IEEE.std_logic_1164;
Entity half_adder is
Port (
        in1, in2: in std_logic;
        out1, out2    : out std_logic
);
End Entity half_adder;

Architecture structure of half_adder is
        component xor_gate is
        port(
                A, B: in std_logic;
                C     :out srd_logic
        );
        End Component xor_gate
```

آزمایش اول

```
        component and_gate is
        port(
                A, B: in std_logic;
                C     :out srd_logic
        );
        End Component and_gate
Begin
        xor_gate_instance0: xor_gate port map (A=>in1, B=>in2, C=>out1);
        and_gate_instance0: and_gate port map (A=>in1, B=>in2, C=>out2);
End structure;
```