Lecture 3

Technology: A manner of accomplishing a task, especially using technical processes, methods, or knowledge.

. Processor Technology

1. General Purpose Processor (Software)

. The GPP designer builds a programmable device which is suitable for many applications to maximize the number of devices sold.

. A GPP has:

a. A program memory

b. A general datapath

Two different people !!

. To an embedded system designer (you), implementing on a GPP is considered the "Software" portion, Simply because the designer programs the processor's memory without worrying about the design of the GPP's hardware.

Pros:

1. Low Time-to-Market
2. Low NRE cost
3. High flexibility
4. Unit cost is low for small quantities
5. High performance for some applications

Cons:

1. Relatively high unit cost for large quantities
2. Lower performance for certain applications
3. Large size
4. Large power consumption

Ali Nawab

2. Single Purpose Processor (Hardware)

- The digital circuit is designed to execute one program (not programmable).
- An SPP doesn't have a program memory and it has a custom datapath.
- The embedded system designer (you) is the one designing the processor which is considered the "Hardware" portion.

Pros:

1. High performance.
2. Small size.
3. Low power consumption.
4. Low unit cost for large quantities.

Cons:

1. High NRE cost.
2. Long design time.
3. Low flexibility.
4. High unit cost for small quantities.
5. GPP performance could be better for some applications.

Ali Nawab

3. Application-Specific Processors

- ASIP: Application-Specific Instruction-Set Processor
- A compromise between GPP & SPP.
- Programmable but optimized for a certain class of applications.
- It has:
  a. A program memory.
  b. A custom datapath.
- An embedded system designer designs a custom datapath (hardware) and programs the processor with the instruction set specific for that processor (Software)
- The instruction set is based on the instructio set architecture (ISA) made by the designer.

Pros:

1. Flexibility.
2. Good performance.
3. Small size.
4. Low power consumption.

Cons:

1. High NRE cost.
2. Long design time.

Examples on ASIPs:

1. Microcontrollers
2. Digital Signal Processors (DSP)

Ali Nawab

. IC Technology

1. Full-Custom / VLSI

- . All layers are optimized.
- . High NRE cost.
- . Long turnaround times.
- . High performance.
- . Small size & power consumption.
- . Used only in high volume or extremely performance-critical applications.

2. Semi-custom / ASIC

- . Lower layers are fully or partially built.
- . Designer optimizes upper layers.
- . In Gate Array ASIC: masks for transistor and gate levels are pre built, the designer only connects the gates.
- . In Standard Cell ASIC: logic-level cells are predesigned but need to be arranged into complete masks and then connected.
- . Good performance & size.
- . Lower NRE cost than full-custom.
- . Require weeks or months to manufacture.

3. Programmable Logic Devices (PLD)

- . All layers already exist.
- . Programmed by creating or destroying connections between wires that connect gates.
- . Very low NRE cost but more power consumption & unit cost.
- . Slow compared to Full Custom & ASIC.

Ali Nawab

.Design Technology

1. Compilation / Synthesis

- Takes a high-level language and generates a lower-level implementation automatically.
- Logic synthesis converts boolean expressions into a netlist.
- Register-transfer synthesis converts FSMs & register transfers into a DP & a CU.
- Software compiler converts a sequintial program into assembly code.
- System synthesis converts an abstract system specification into a set of sequential programs on processors.

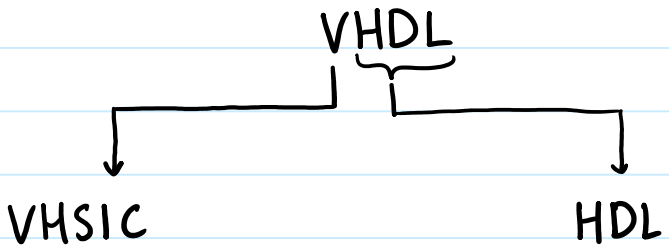
2. Libraries /IP

- Libraries: The reuse of preexisting implementations.
- IP (Intellecual Property): Using libraries of cores implementing portions of ICs, these cores are protected from copying.

3. Test / Verification

- Testing: Ensuring the correctness of the system's functionality by giving samples of inputs and observing outputs, simulation is usually used for testing.
- Verification: Formal verification (mathematical).

Ali Nawab



Very High Speed Integrated Circuit Hardware Descriptive Language

Assembly Language: MIPS, PIC, x86
High Level Language: Java, C++, Swift
Hardware Descriptive Language: VHDL, Verilog

Descriptions:

1. Behavioral Description

Describes the behavior, function, or dataflow of the circuit.

a. Algorithms → Process

sequential
if, else, loop

b. Dataflow → Built-in operators

2. Structural Description

Describes how components are inter-connected in the circuit.

Port maps

Ali Nawab

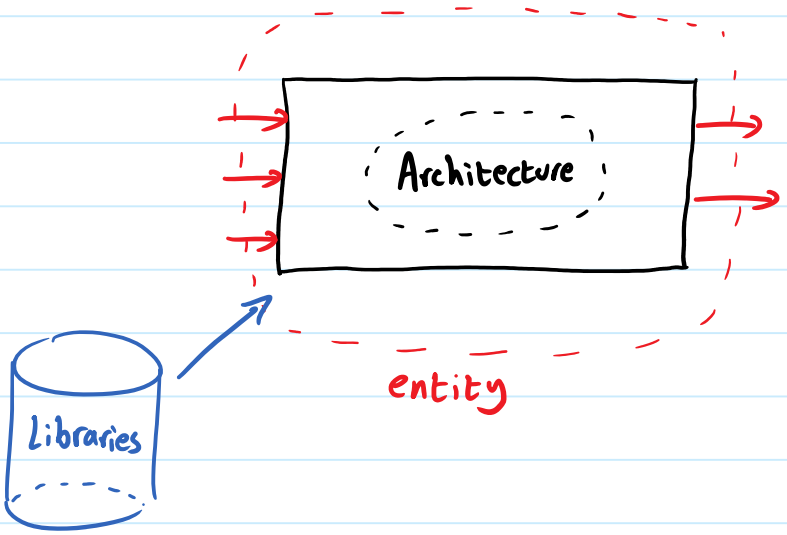
A VHDL file consists of 3 main parts:

[1] Libraries: Pre existing code that we just use (IEEE libraries).

[2] Entity: Describes the inputs and outputs of the circuit.

[3] Architecture: Describes the inside of the circuit (Structure and/or Behavior).

* Note that there is more to VHDL than just these parts, moreover, each part could have more features than what is stated in the definitions, however, the information here is all we need for upcoming material.



Ali Nawab

VHDL Template

```
library ieee;
use ieee.std_logic_1164.all;

entity name is
port(
  pin0, pin1: mode type;
  pin2: mode type;
  pin3, pin4: mode type
);
end entity;

architecture behavior/structure of name is

  -- Declarations
    1. variables, constants, signals
    2. functions
    3. components
    4. new types (states, arrays)

begin

  -- Circuit Description (the inside)
    1. Port maps (Structural)
    2. Process (Behavioral)
    3. Dataflow (Behavioral)

end architecture;
```

```
// Java comment
-- VHDL comment
```

mode:

1. in → input
2. out → output
3. inout → input or output
4. buffer → An output whose value can be taken as input

Type:

1. bit: 0, 1
2. std_logic: 0, 1, w, h, l, U, z, x, -
3. std_logic_vector (range)
   downto (7 downto 0)
   to (0 to 7)
4. Signed (range)
5. unsigned (range)
6. real
   ↳ cannot be synthesized!!
   (for simulation only)

Ali Nawab