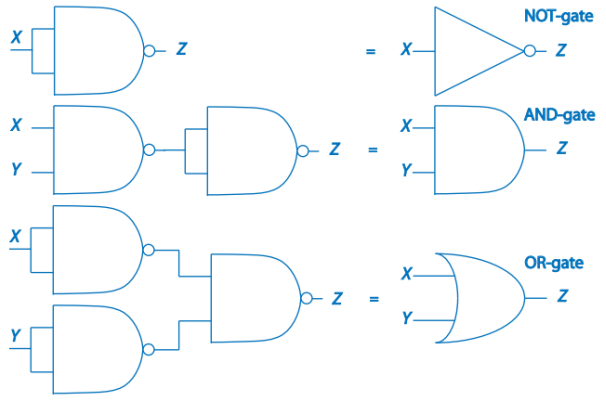
What are the three fundamental phases of integrated circuit design? You should briefly explain what each phase does. [3]

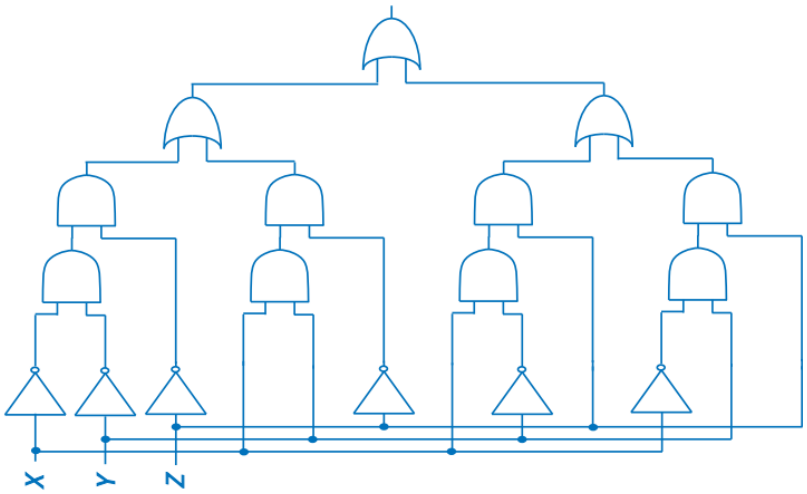
What is a NAND-gate? Show how a NOT-gate, an AND-gate and an OR-gate can be constructed using just NAND-gates. [10]

Build a circuit using NOT-, AND- and OR-gates that computes the function with 3 Boolean inputs and where the output is 1 if, and only if, there are an even number of 1's in the input. [6]

A **functional specification** [$\frac{1}{2}$] of the chip is derived (this describes exactly what the IC is supposed to do involving factors like chip area, power, speed and cost [$\frac{1}{2}$]); the **register transfer level (RTL)** [$\frac{1}{2}$] design is then undertaken (the functional specification is used to describe the exact behaviour of the digital circuits on the chip, as well as the interconnections to inputs and outputs, with this description being in terms of logic gates and interconnecting wires [$\frac{1}{2}$]); and then the RTL design is **mapped to a physical layout** [$\frac{1}{2}$] in silicon (specific physical attributes must be respected; for example, it is crucial that appropriate spacing between transistors is maintained in the physical layout [$\frac{1}{2}$]).

A NAND-gate takes two inputs and outputs 0 if, and only if, both inputs are 1 [1]. [2] for NOT-gate; [3] for AND-gate; [4] for OR-gate



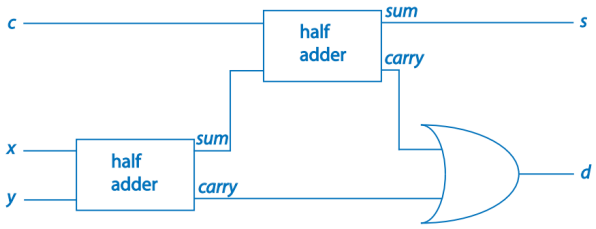


What are a half-adder and a full-adder? Show how a full-adder can be built using 2 half-adders. [7]

Name three key components in a CPU microarchitecture and briefly describe the purpose of each. [3]

Name two types of bus within a CPU and explain the general purpose of each. What is the width of a bus? Explain how the width of a bus imposes memory or data limitations within a CPU. [6]

A half-adder takes x and y as inputs and computes the Boolean sum of x and y, where the Boolean sum of a collection of inputs is 1 if, and only if, an odd number of the inputs are 1, and also the resulting carry bit, which is 1 if, and only if, both x and y are 1 [2]. A full-adder takes x, y, and z as inputs and computes the Boolean sum of x, y, and z, resulting in the sum-bit and the carry-bit [2].



The **datapath** [$\frac{1}{2}$] performs all the data processing operations and includes the arithmetic logic unit (ALU), which is the part of the CPU that performs all the arithmetic and logical operations on data, and a limited number of memory locations called registers [$\frac{1}{2}$]. The **control** [$\frac{1}{2}$] tells the datapath, memory and input/output (I/O) devices what to do (it is the conduit between the datapath and the main memory) [$\frac{1}{2}$]. A **cache** [$\frac{1}{2}$] consists of small, fast and relatively expensive on-chip memory and is used to store memory items that need to be regularly accessed [$\frac{1}{2}$].

Buses include a **data bus** [$\frac{1}{2}$], an **address bus** [$\frac{1}{2}$] and a **control bus** [$\frac{1}{2}$]. A data bus carries the contents of memory locations between the processor and main memory [$\frac{1}{2}$]. The address bus holds addresses of locations in main memory [$\frac{1}{2}$]. The control bus is used to transfer information between the CPU and various other devices within the processor [$\frac{1}{2}$]. The width of a bus is the number of parallel wires in a bus [1]. The width of the data bus determines the word-size of the computer [1]. The width of an address bus determines the size of addressable memory [1].

Explain carefully the four phases of the fetch-decode-fetch-execute processor cycle (be sure to explain the purpose of any CPU components you happen to mention). [5]

Show different layers of abstraction in a modern computer system. [4]

The '**instruction fetch**' phase involves the supply of the instruction address (via the address bus) and the return from memory (via the data bus) of the instruction [1]. The '**instruction decode**' phase involves interpreting the stored instruction within the CPU [1]. The '**operand fetch**' phase involves the supply of the address of any required data (via the address bus) and the return from memory (via the data bus) of this data [1]. The '**execute instruction**' phase involves the CPU performing the necessary actions [1] (this phase is sometimes split into two with the 'execute instruction' phase followed by a 'write-back' phase where data is written back to memory, if needs be [1]).