The basic HW design flow introduction

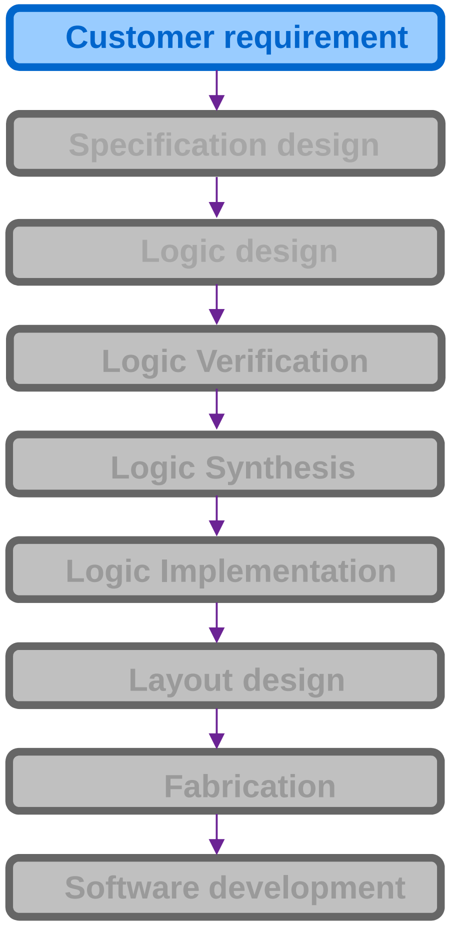
The basic HW design flow introduction

1. Customer requirement *2. Specification Design* *3. Logic Design* *4. Logic Verification* *5. Logic Synthesis* *6. Logic Implementation* *7. Layout Design* *8. Fabrication* *9. Software Development*

To product one chip, basically, we need to process through 9 ***steps***:

- *Step 1* is processed by the Marketing or Selling department.
- *Step 2, 3, 4, and 5* are processed by the Design department and are called Frontend Design.
- *Step 6 and 7* are processed by the Design department and are called Backtend Design.
- *Step 8* is processed by the Manufacture department.
- *Step 9* is processed by the Software department.

The basic HW design flow introduction

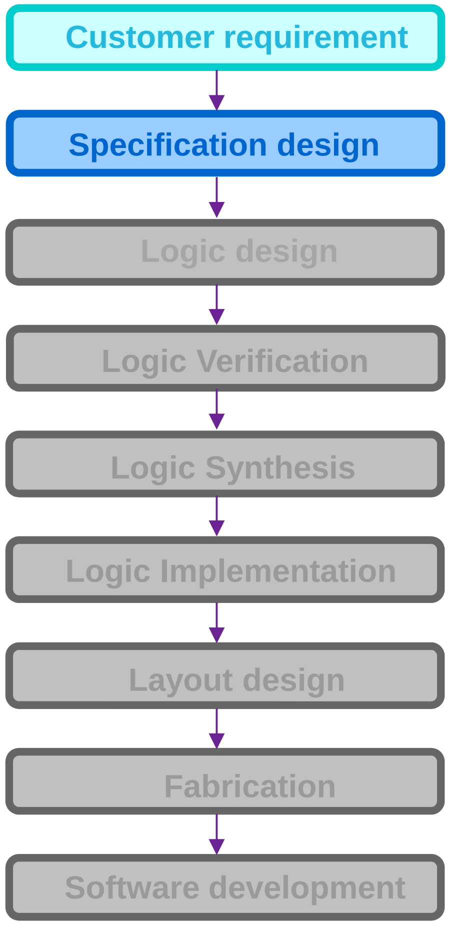


Step 1: Customer Requirement

Marketers or sellers, along with engineers, will contact with customers to negotiate about demands or predicts market's needs.

- The requirements will be written in *a contract* by *common language.*
- Customers come from America, Europe and Asia's companies
- Our main products: MPU, MCU, SoCs for mobile phones, car information systems ....

The basic HW design flow introduction

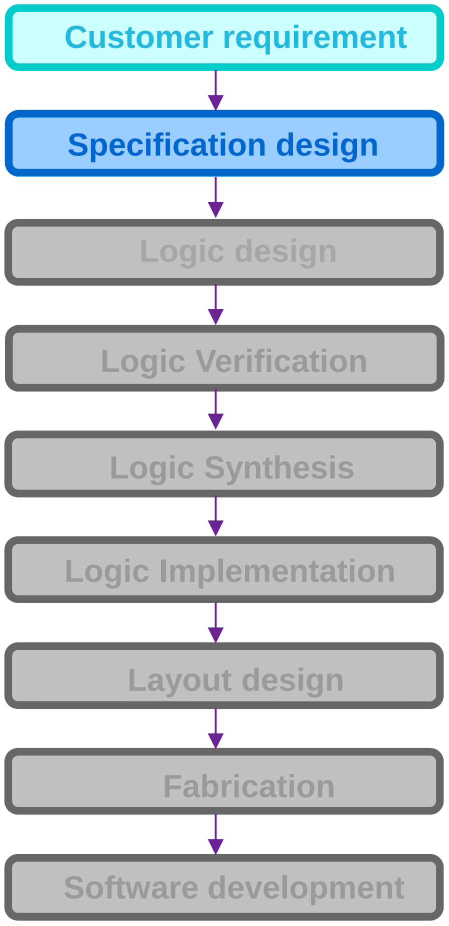


Step 2: Specification Design [Processor, IP, SoC, FE]

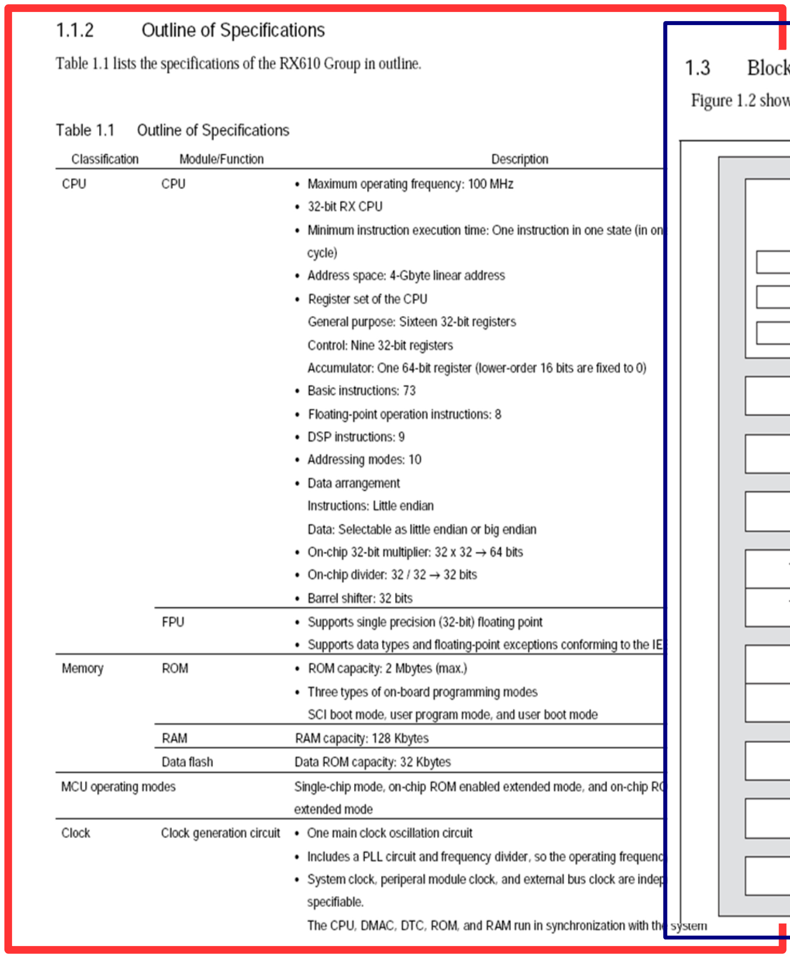
The demands of customers will be transfered into *technique documents (specifications)*.

- Speed of the chip : 1 Ghz, 200 Mhz, 100 Khz ...
- How much power consumption of the chip
- How many implemented peripherals
- How many CPU cores
- How to design and verify this chip ...
- ....

The basic HW design flow introduction



Step 2: Specification Design



1.1.2 Outline of Specifications

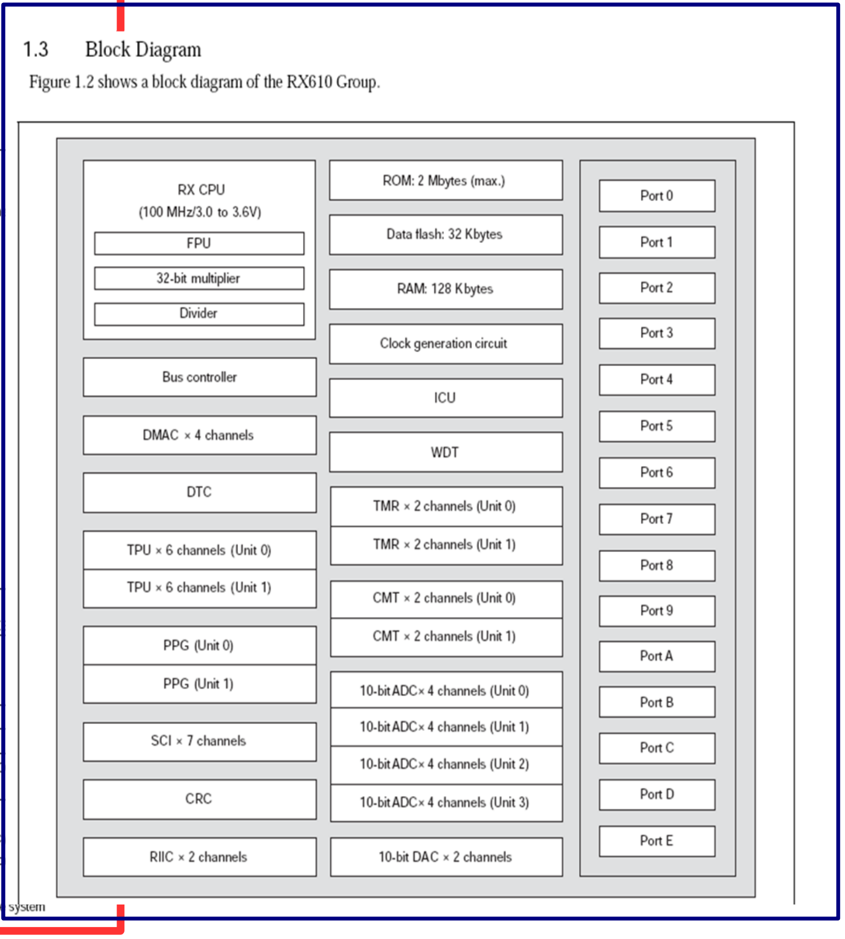
Table 1.1 lists the specifications of the RX610 Group in outline.

Table 1.1 Outline of Specifications

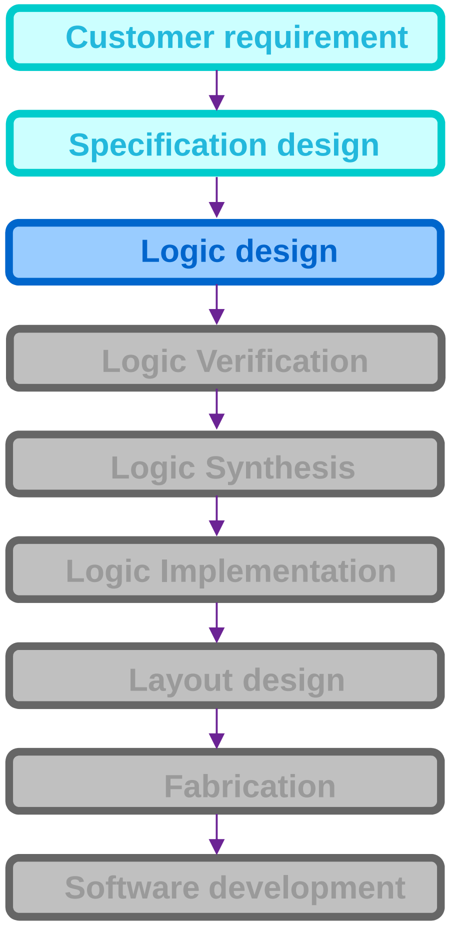
Classification	Module/Function	Description
CPU	CPU	• Maximum operating frequency: 100 MHz • 32-bit RX CPU • Minimum instruction execution time: One instruction in one state (in on cycle) • Address space: 4-Gbyte linear address • Register set of the CPU General purpose: Sixteen 32-bit registers Control: Nine 32-bit registers Accumulator: One 64-bit register (lower-order 16 bits are fixed to 0) • Basic instructions: 73 • Floating-point operation instructions: 8 • DSP instructions: 9 • Addressing modes: 10 • Data arrangement Instructions: Little endian Data: Selectable as little endian or big endian • On-chip 32-bit multiplier: 32 x 32 → 64 bits • On-chip divider: 32 / 32 → 32 bits • Barrel shifter: 32 bits
	FPU	• Supports single precision (32-bit) floating point • Supports data types and floating-point exceptions conforming to the IE
Memory	ROM	• ROM capacity: 2 Mbytes (max.) • Three types of on-board programming modes SCI boot mode, user program mode, and user boot mode
	RAM	RAM capacity: 128 Kbytes
	Data flash	Data ROM capacity: 32 Kbytes
MCU operating modes		Single-chip mode, on-chip ROM enabled extended mode, and on-chip RO extended mode
Clock	Clock generation circuit	• One main clock oscillation circuit • Includes a PLL circuit and frequency divider, so the operating frequenc • System clock, periperal module clock, and external bus clock are indep specifiable. The CPU, DMAC, DTC, ROM, and RAM run in synchronization with the system

1.3 Block Diagram

Figure 1.2 shows a block diagram of the RX610 Group.



The basic HW design flow introduction

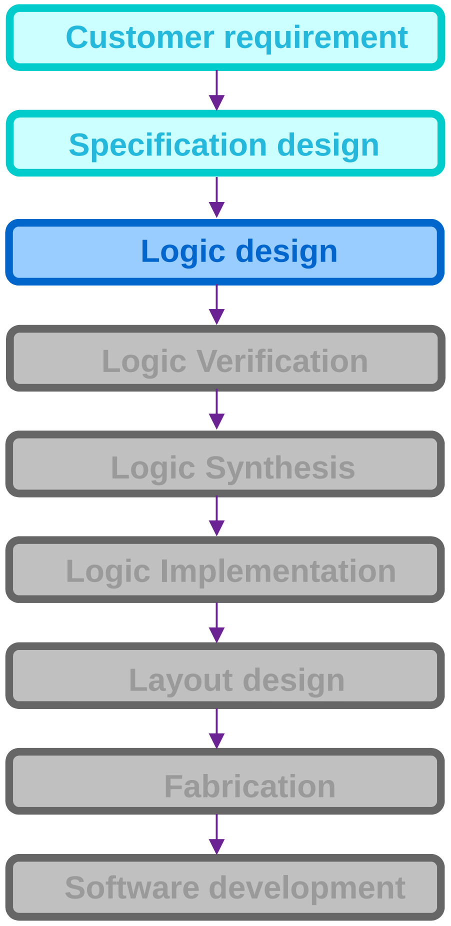


Step 3: Logic Design [Processor, IP, SoC, FE]

Logic Design or RTL (Register Transfer Level) Design is a step in which chip's specifications is designed by using HDL
(Hardware Design Language)

- Two commons HDL languages are Verilog and VHDL
  - *Verilog: Verification Logic [Verilog 1995, 2001, System Verilog]*
  - *VHDL: VHSIC (Very High Speed Integrated Circuit) Hardware Design Language*
- Today, Verilog is the most use in Hardware Design

The basic HW design flow introduction



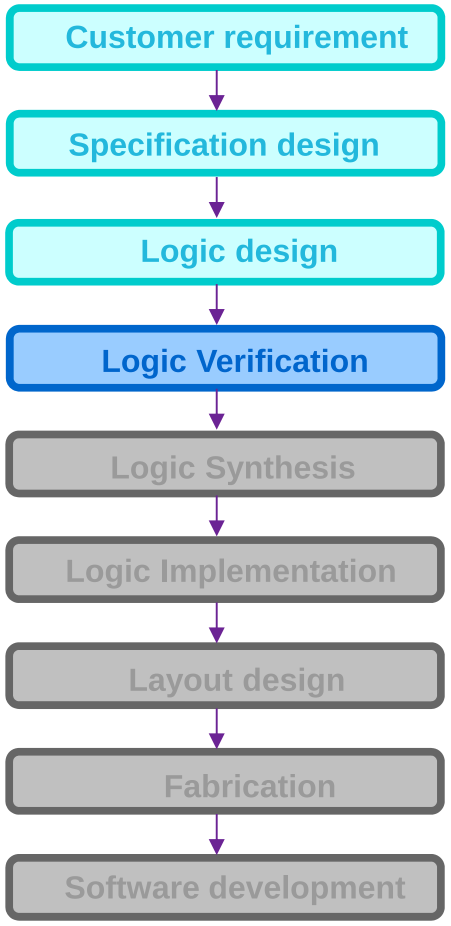
Step 3 : Logic Design [Processor, IP, SoC, FE]

```
// A flip-flop
module ff (clk,rst_n,in,out);
   input clk;     // clock signal
   input rst_n;   // reset signal
   input in;      // data input signal (1 bit)

   output out;    // data output signal (1 bit)
   reg out;       // register contains output data

   // Reset data when rst_n is 0
   // Update data for each rising clock when
   // rst_n is 1
   always @(posedge clk) begin
      if (rst_n == 1'b0) begin
         out <= 1'b0;
      end
      else begin
         out <= in;
      end
   end
endmodule
```

The basic HW design flow introduction

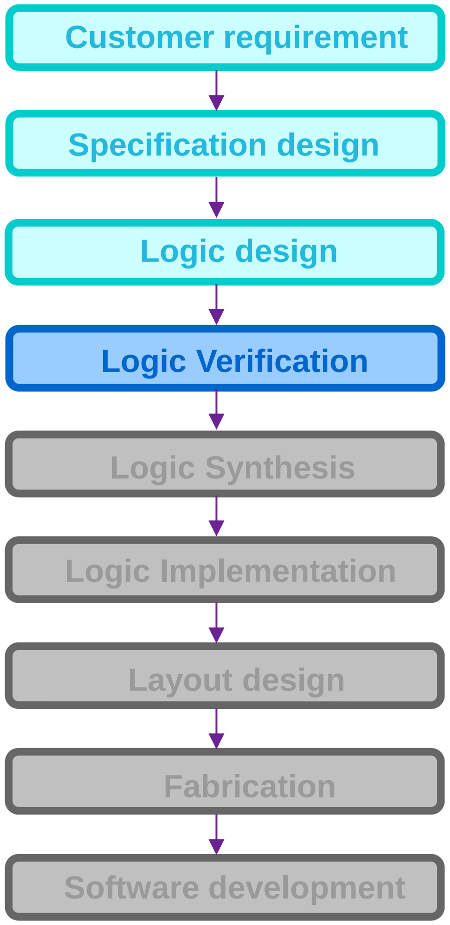


Step 4: Logic Verification [Processor, IP, SoC, FE]

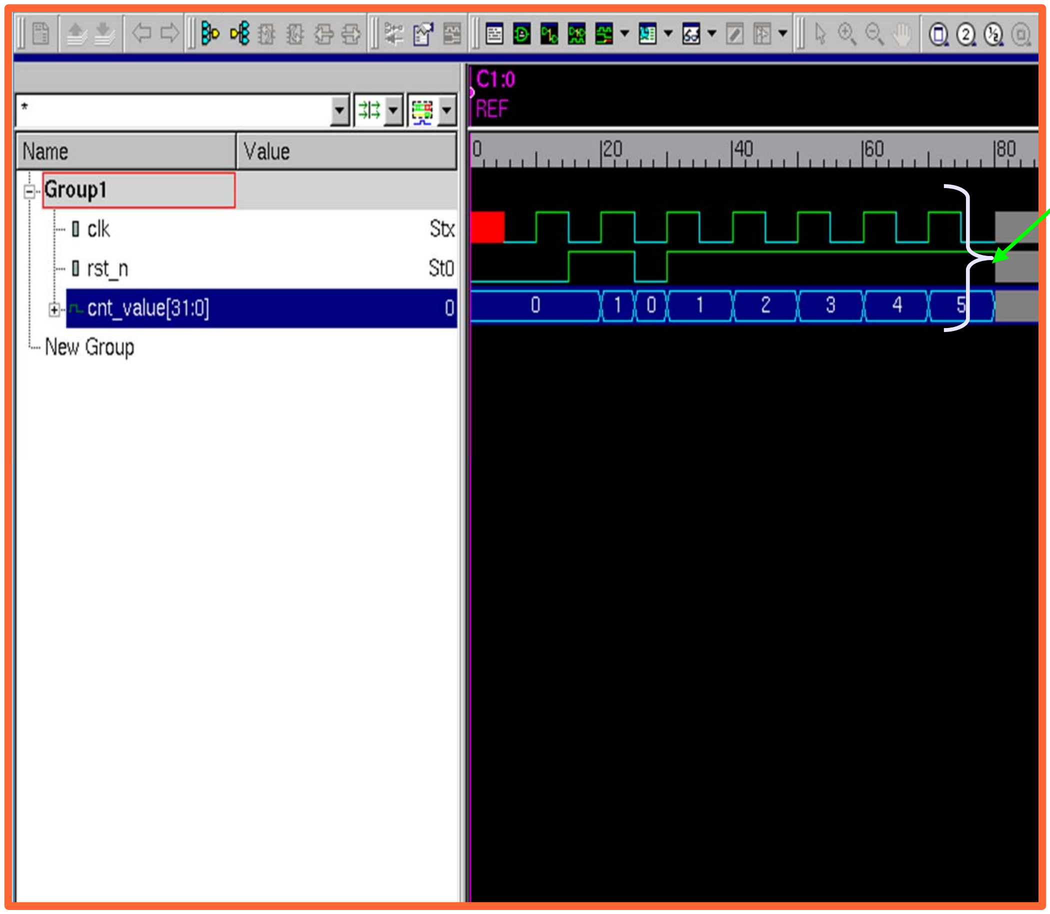
This step is used to execute (simulate) RTL code on a simulator.

- To check whether functions of the RTL run correct or not.
- To check whether the timing (relationships among signals) of the chip run correct or not.

The basic design flow introduction

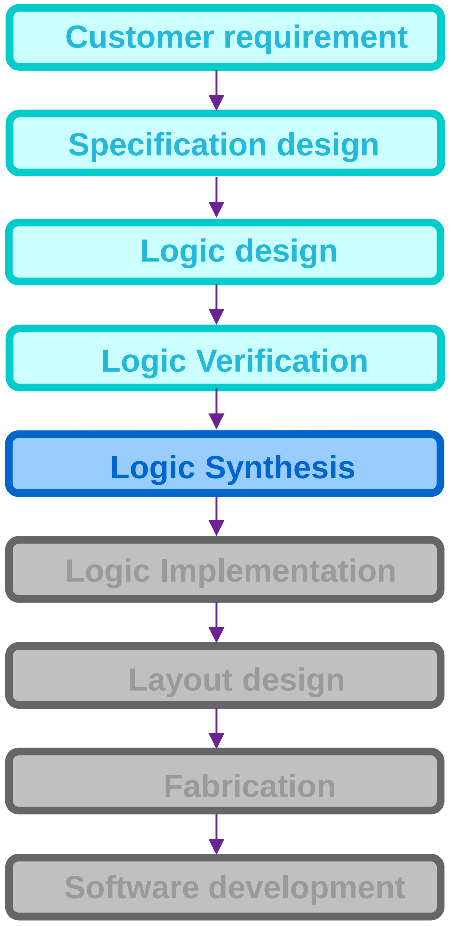


Step 4: Logic Verification [Processor, IP, SoC, FE]



+ This is an example about a waveform.
+ This waveform shows a relationship among three signals : clock signal, reset signal and output signal

The basic design flow introduction



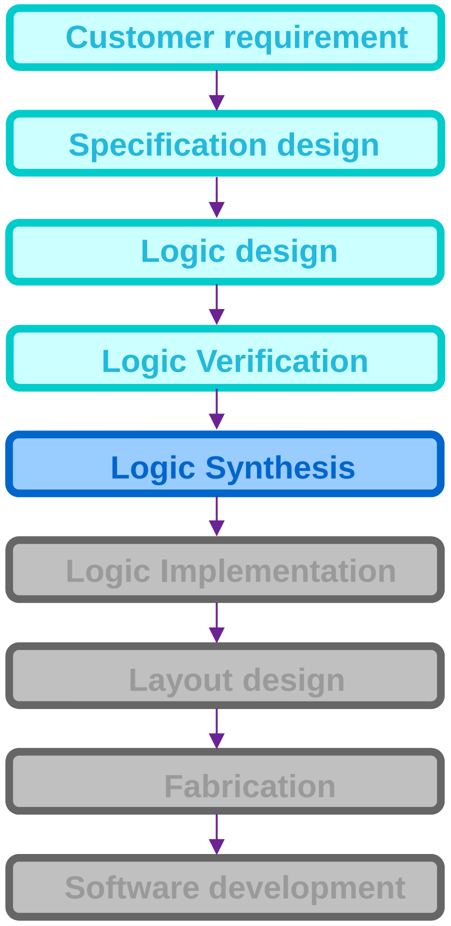
Step 5: Logic Synthesis [Processor, IP, SoC]

This step will transfer a RTL code into a gate net list

- The advantage of each production process will be displayed here. Each cell will have different characteristics such as area, power consumption ... according to a corresponding process such as 60 nm, 45 nm, 28 nm and etc.

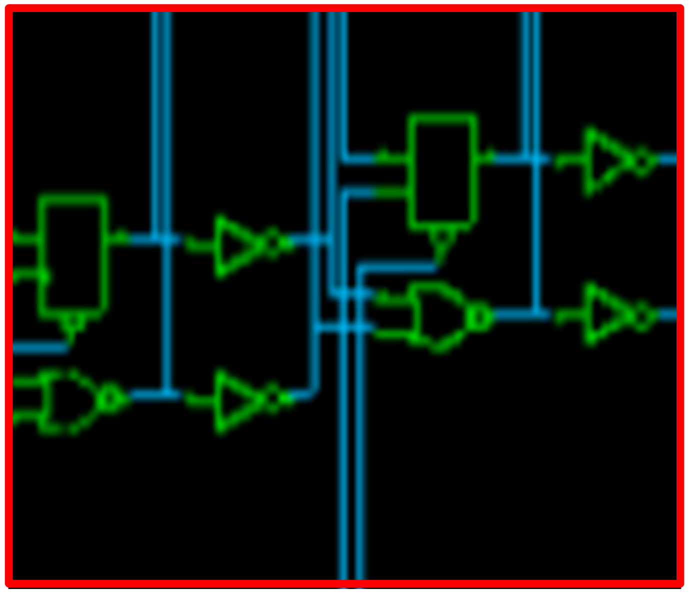
- The smaller synthesized area, the more advantages a chip gains.

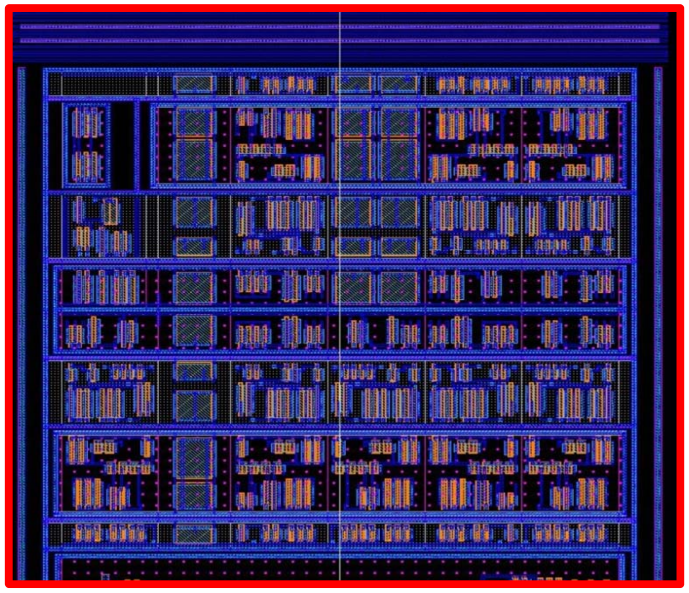
The basic design flow introduction



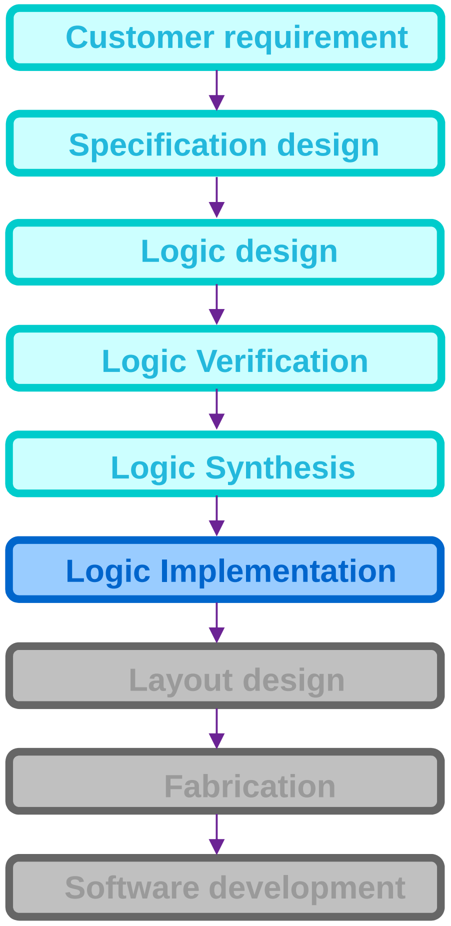
Step 5: Logic Synthesis

[Processor, IP, SoC, Circuit Design]


A gate net list


A part of the standard cell

The basic design flow introduction



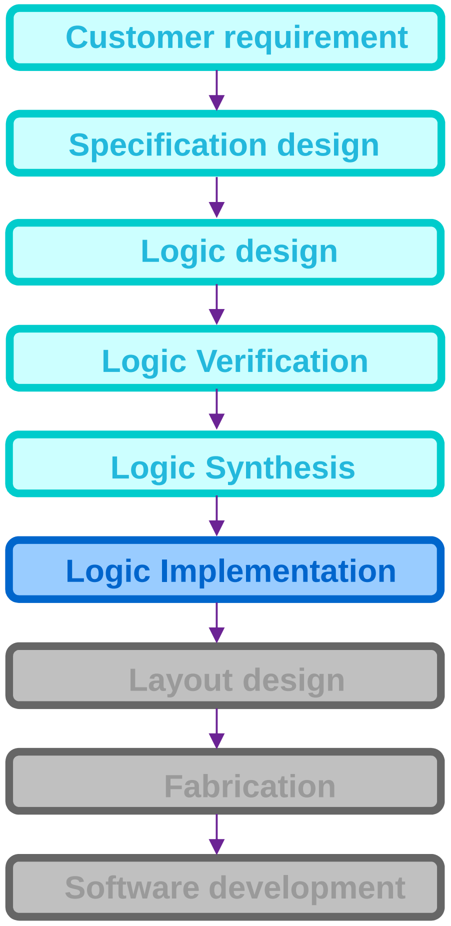
Step 6: Logic Implementation

[Logic Implementation, Processor, FE]

This step will work mainly on a gate net list for:

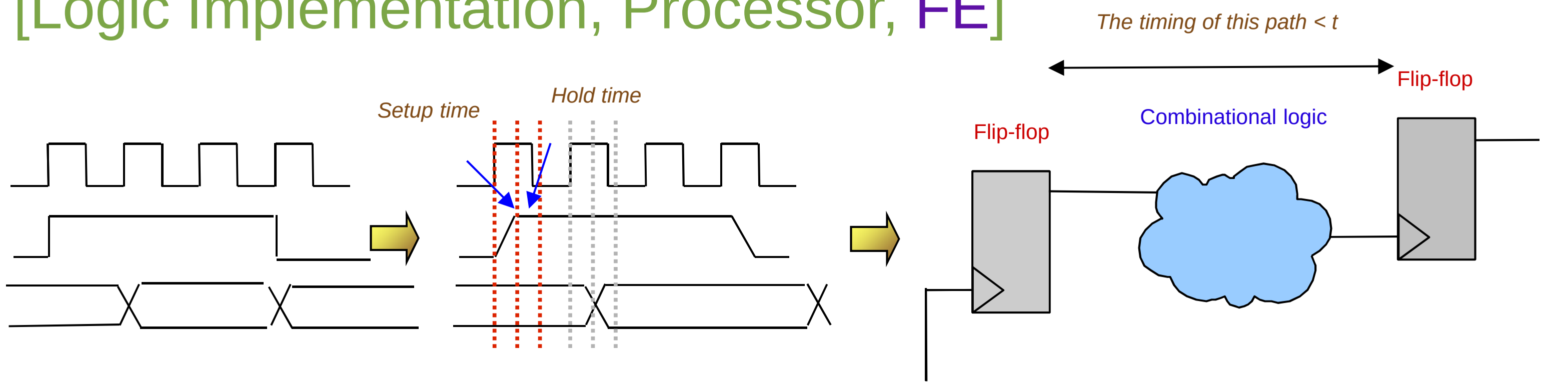
- Check timing of a chip in such field : hold/setup time, critical paths ...
- Add an extra gate net list for the DFT (Design For Test) function to enhance the later productivity

The basic design flow introduction



Step 6: Logic Implementation

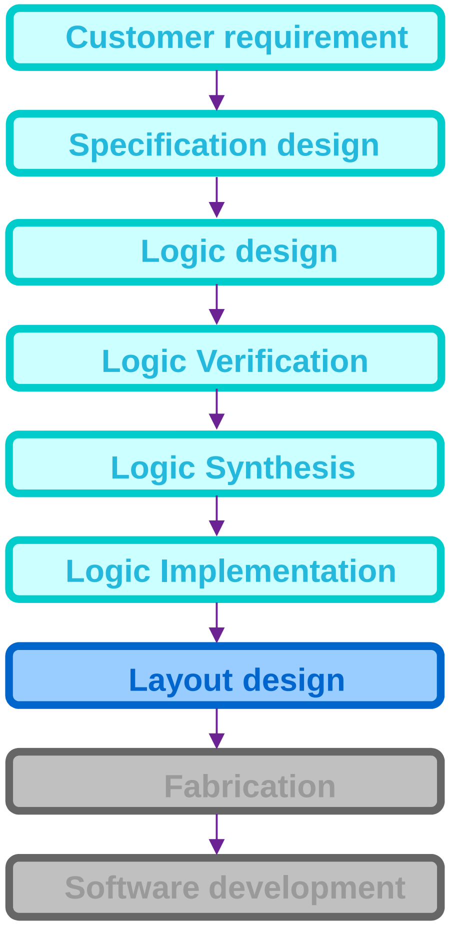
[Logic Implementation, Processor, FE]



DFT: *During manufacturing, if a chip is defected, how do we diagnose the position of the error?*

Add some circuits will help designers to find the correct position and fix the errors in some cases

The basic design flow introduction

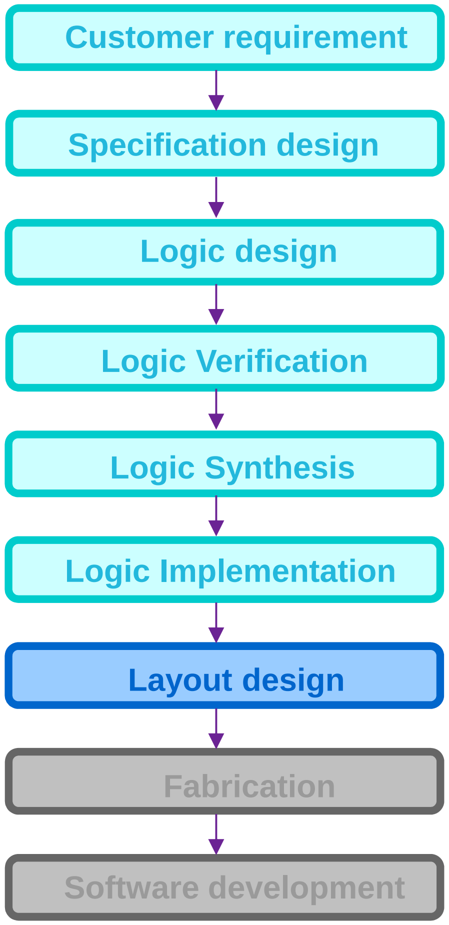


Step 7: Layout Design [Backend]

This step will work arrange the chip's gate net list into a specified area (wafer).

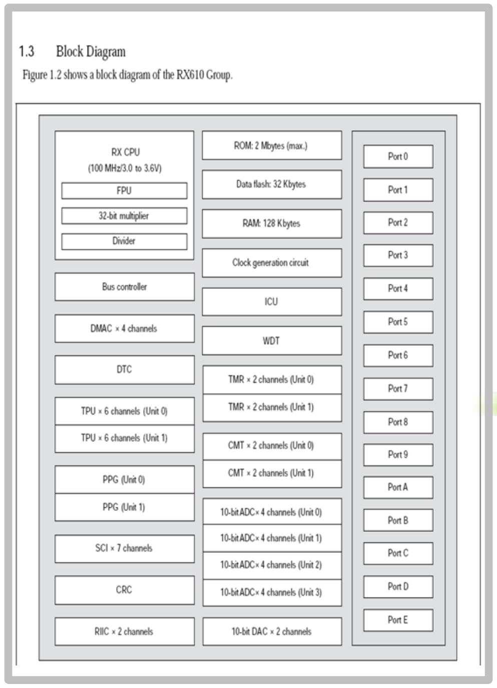
- Minimize the chip's area.
- Check physical conditions can affect to the chip during the arrangement process.
- Arrange power supply for a chip.
- Wiring signals, etc.

The basic design flow introduction



Step 7: Layout Design [Backend]

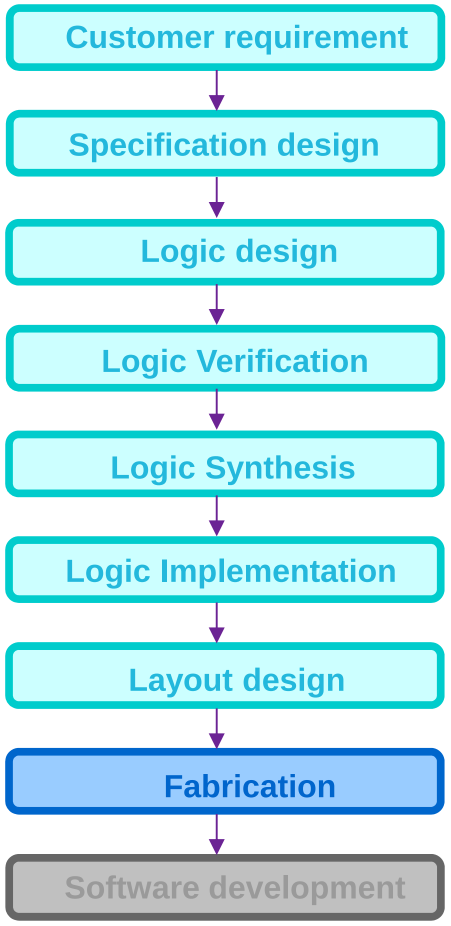
Each color is a module
A layout designer must find a way to arrange all module in the smallest area but this chip still works correctly



1.3 Block Diagram

Figure 1.2 shows a block diagram of the RX610 Group.

The basic design flow introduction



Step 8: Fabrication

The chip will be produced and packaged in a manufacture

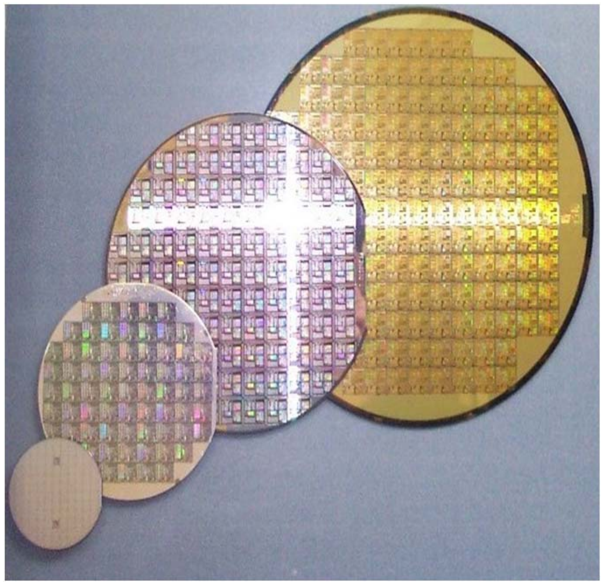
- This step will be done by Japanese factories
- Silicon (wafer) is a main material to product a chip
- A factory will cost from 3 ~ 4 billion to build.
- To product each manufacture process, a factory must be equipped with corresponding machines

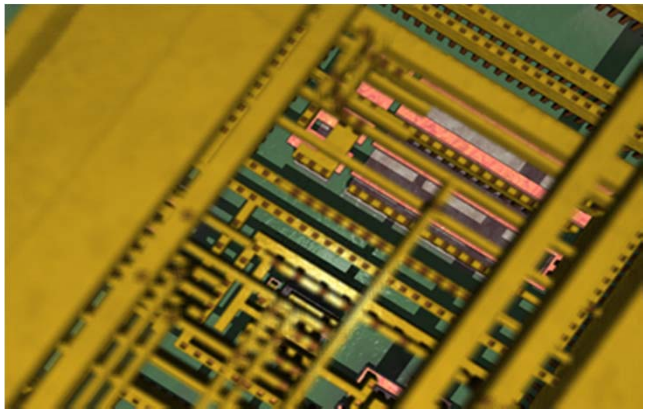
The basic design flow introduction

- Customer requirement
- Specification design
- Logic design
- Logic Verification
- Logic Synthesis
- Logic Implementation
- Layout design
- **Fabrication**
- Software development

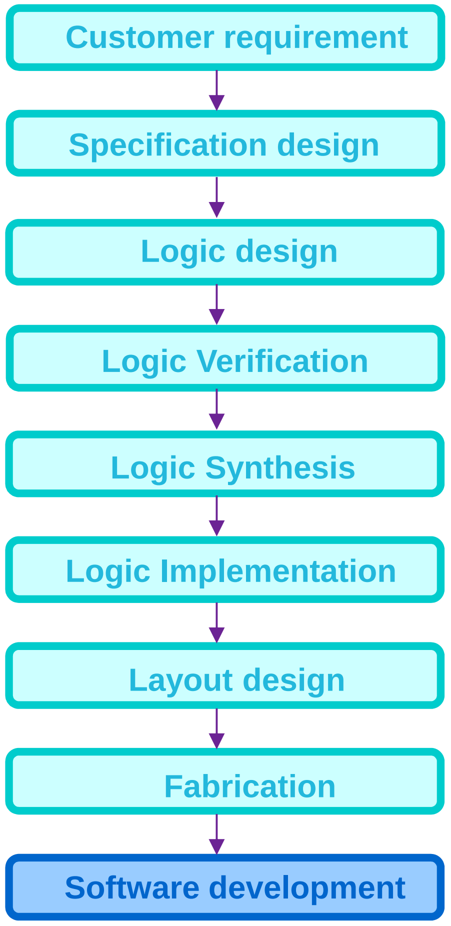
Step 8: Fabrication

Clean room to product chip


Wafer's sizes


A chip on a wafer

The basic design flow introduction



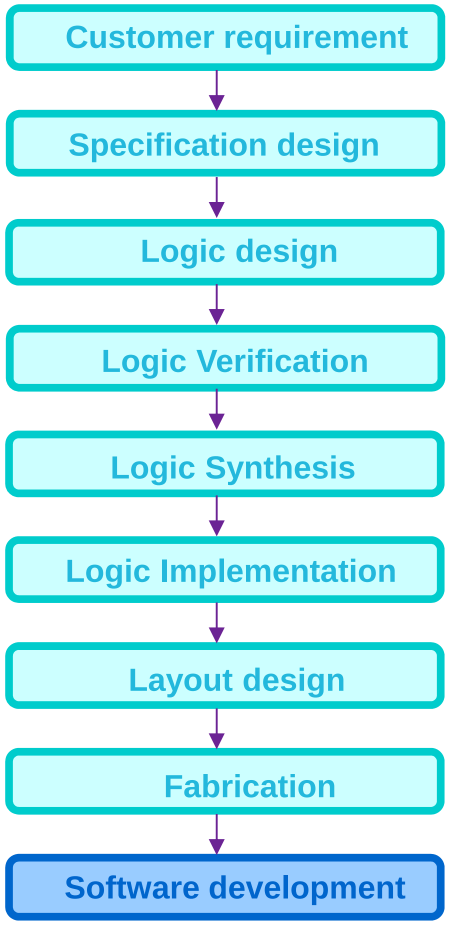
Step 9: Software development

[Software solution, Mobile platform, Software IP Processor, IP, and SoC]

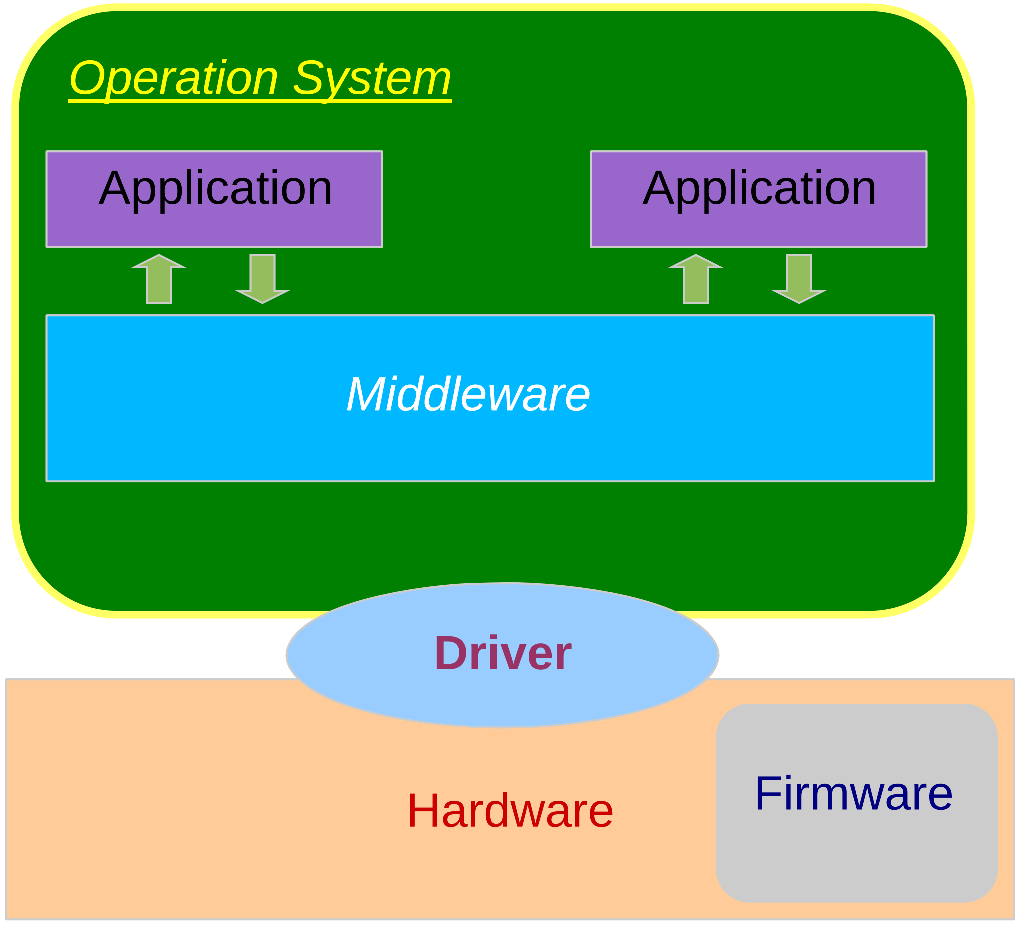
This is the final step in developing a chip. Renesas will also provide a complete or, part of, software solution for customers through:

- Port a operation system on chip.
- Develop demonstration, application, firmwares, middle wares and drivers.
- Develop tool kits for third parties.

The basic design flow introduction



Step 9: Software development



Operation System:
manage hardware resources and provide efficient executions for applications
Application:
designed to help the user to perform singular or multiple related specific tasks
Middleware:
help to provide the communication among applications
Driver:
help operation system to manage operate hardware resources
Firmware:
integrated in a hardware to perform some particular functions.