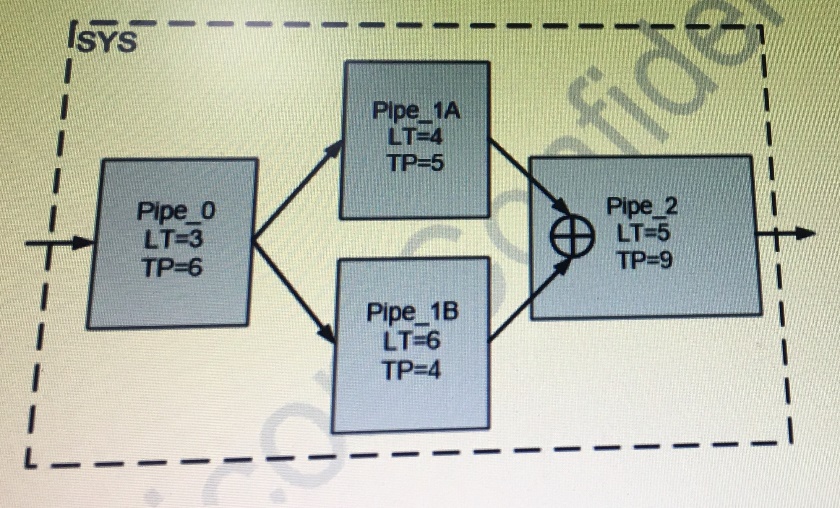
**17、Please describe the definition of throughput and latency. Please also calculate out the throughput and latency value of below pipeline system and show the detail step of calculation process. (10分)**

**请描述吞吐量和延迟的定义。还请计算出流水线下系统的吞吐量和延迟值，并给出计算过程的详细步骤。（10分）**

Assumption: Pipe\_1A needs to process all data of pipe\_0; Pipe 1B needs to process all data of pipe\_0 too; Pipe\_2 can only work when data from Pipe\_1A and Pipe\_1B both arrive.

LT- stands for latency which is in unit of cycles.

TP- stands for throughput which is in unit of bits/cycles.



**18、In below design, the delay assumptions are as:**

1) Clock uncertainty = 0.3 ns;

2) FF Cell delay = 2 ns;

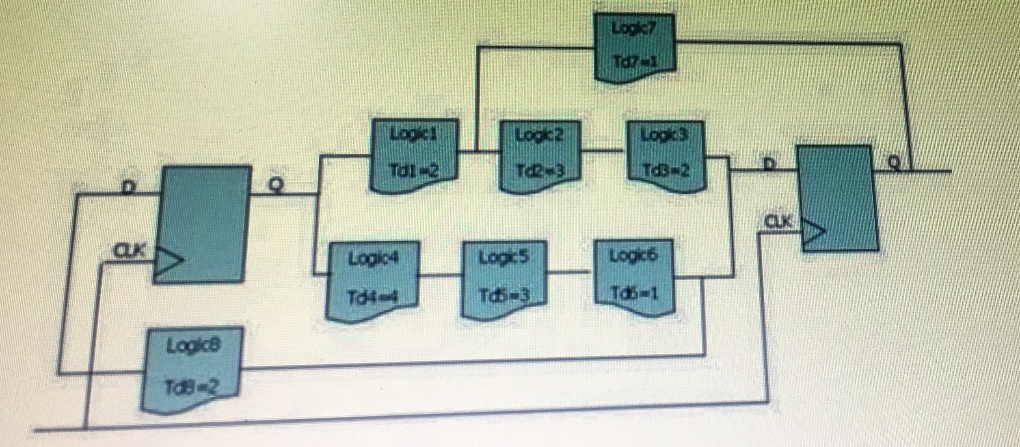
3) FF setup = 0.2 ns;

4) FF hold = 0.1 ns;

Please do:

a).Draw setup and hold critical path in picture.(For online tests, question a can be skipped)

b).Calculate maximum clock frequency for the design, and hold slack Thold.



解：

Tcycle + Tskew > Tco + Tcomb + Tsu

Tcycle > Tco + Tcomb + Tsu – Tskew

Tcycle > 2ns + 8ns + 0.2ns – 0.3ns = 9.9ns

因此电路最大时钟频率为1s/9.9ns ≈ 100MHz

**19、what are the differences between “virtual task/function” and “task/function” in Systemverilog? Is it ok if I declare new function as virtual?**

**20、What is cache? How dose it work?**

**21、Time division multiplexing technology (TDM) can combine multiple identical logic modules into one logic module by increasing the clock frequency, thus reducing the resource consumption in FPGA.**

**Please follow these steps to complete a simple TDM design and write your RTL code. (10分)**

**时分复用技术（TDM）可以通过提高时钟频率将多个相同的逻辑模块组合成一个逻辑模块，从而降低FPGA的资源消耗。**

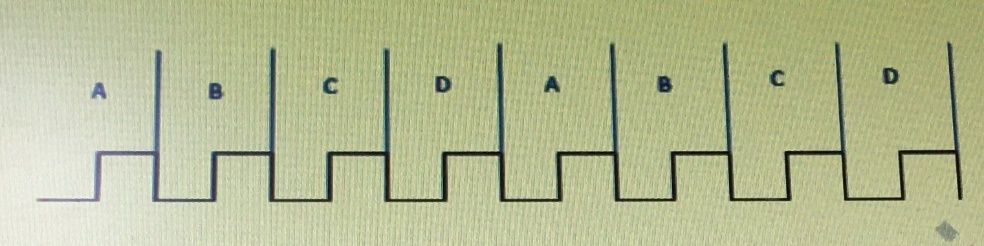
**请按照以下步骤完成一个简单的TDM设计并编写RTL代码。（10分）**

1) The shift register can delay the output of the input signal. Please design a shift register to delay the output for 3 cycles after the input signal is locked.

2) As shown in the figure, divide the clock cycle into 4 region and repeat. There are enable signal EN and input signal DATA\_IN[3:0].

1） 移位寄存器可以延迟输入信号的输出。请设计一个移位寄存器，在输入信号被锁定后将输出延迟3个周期。

2） 如图所示，将时钟周期分为4个区域重复。使能信号EN和输入数据DATA\_IN[3:0]。



At clock rise edge in region A, a register latch DATA\_IN[0] when EN asserted high, otherwise it keep the last enable DATA\_IN[0].

At clock rise edge in region B, a register latch DATA\_IN[1) when EN asserted high, otherwise it keep the last enable DATA\_IN[1].

For region C and D, they have same logical relationship to latch DATA\_IN[2] and

DATA\_IN[3] as above. Please use shift register to achieve the function.

3) There is a combined logic module, its top-level interface is:

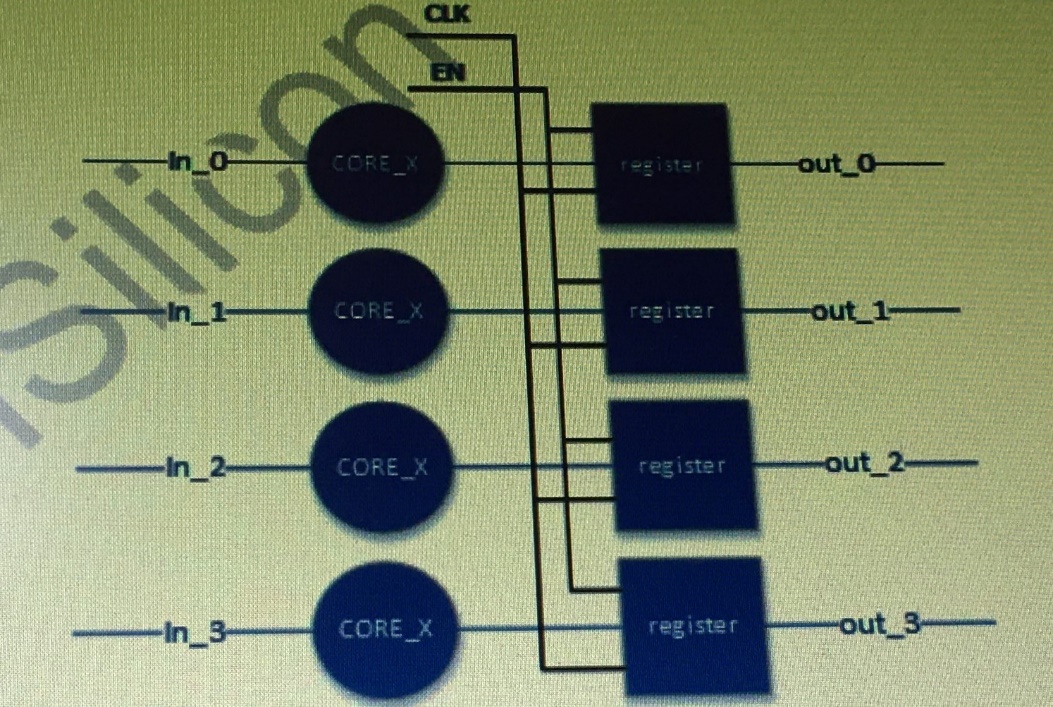
module CORE\_X(

input [15:0] in,

output [15:0] out

);

There is a project used 4 CORE\_X as follow:



Now we increase the clock frequency by 4 times. Please use a shift register to implements the same function by use only 1 CORE\_X.