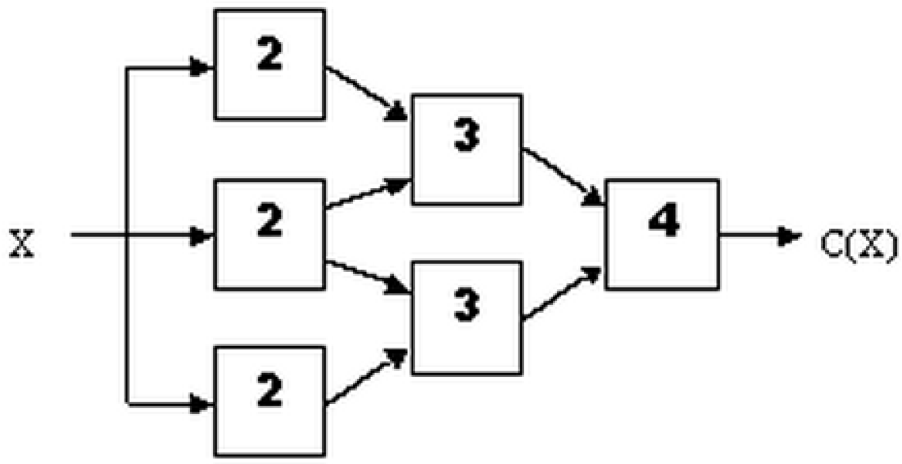
LE7.2.1 Pipeline Design

4/5 points (ungraded)

Consider the following combinational encryption device constructed from six modules:



The device takes an integer value, X, and computes an encrypted version C(X). In the diagram above, each combinational component is marked with its propagation delay in ns; contamination delays are zero for each component.

In answering the following questions assume that registers added to the circuit introduce no additional delays (i.e., the registers have a contamination and propagation delay of zero, as well as zero setup and hold times). Any modifications must result in a circuit that obeys our rules for a well-formed pipeline and that computes the same results as the combinational circuit above. Remember that our pipeline convention requires that every pipeline stage has a register on its output.

(A) What is the latency of the combinational encryption device?

Latency of device (ns): 9 ✔ Answer: 9

Explanation

Latency = delay along longest path from input to output = 2 + 3 + 4 = 9.

(B) If we want to increase the throughput of the encryption device, what is the minimum number of registers we need to add?

Minimum number of registers: 3 ✔ Answer: 3

Explanation

Playing by our pipelining rules, we always add a register to the output. To increase the throughput, we need to add another contour that bisects the circuit. The cheapest place to do this is just before the "4" module, requiring two additional registers.

(C) If we are required to add exactly 5 registers, what is the best throughput we can achieve?

Maximum throughput (1/ns): 1/7 ✖ Answer: 1/5

Explanation
The best throughput we can achieve with 5 registers is 1/5: place 3 (!) registers on the output and two registers on the arcs leading to the "4" module. If we use 4 registers to divide the circuit between the "2" and "3" modules and place one on the output, the resulting throughput is 1/7.

(D) If we can add as many registers as we like, what is the upper bound on the throughput we can achieve?

Maximum throughput (1/ns): 1/4 ✔ Answer: 1/4

Explanation
1/4, because the best we can do by just adding registers is to segregate the "4" module into its own pipeline stage.

(E) If we can add as many registers as we like, what is the lower bound on the latency we can achieve?

Minimum latency (ns): 9 ✔ Answer: 9

Explanation
Lower bound on latency = 9. We can never make the latency less by adding pipeline registers; usually the latency increases.

Submit

Answers are displayed within the problem

LE7.2.2 Pipelining Combinational Logic

2/2 points (ungraded)

A combinational circuit C, built entirely from 2-input NAND gates having a propagation delay of 2 ns, has a propagation delay of 20 ns. You pipeline C for maximum throuput using the minimum number of registers necessary; the registers have 1ns setup time and 1ns propagation delay. What would you expect for the latency of the resulting pipeline?

Latency of pipelined version (ns): 40 ✔ Answer: 40

Throughput of pipelined version (1/ns): 1/4 ✔ Answer: 1/4

Explanation

The problem tells us that the combinational propagation delay is 20 ns for a circuit that consists entirely of NAND gates whose propagation delay is 2 ns each. This means that the longest path through the circuit consists of 10 NAND gates. In order to pipeline the circuit for maximum throughput, we need to split the circuit into 10 pipeline stages each of which contains exactly one of these 10 NAND gates. So our longest path now looks like a NAND gate followed by a pipeline register repeated 10 times. The clock period of this pipelined circuit needs to allow sufficient time for the propagation delay of the pipeline register, the propagation delay of the NAND gate and the setup time of the next register. This means that the clock period = 1 + 2 + 1 = 4 ns and throughput is 1/clock_period = 1/4. The latency of our pipelined circuit is the number of stages times the clock period which is 10 * 4 = 40 ns.

Submit

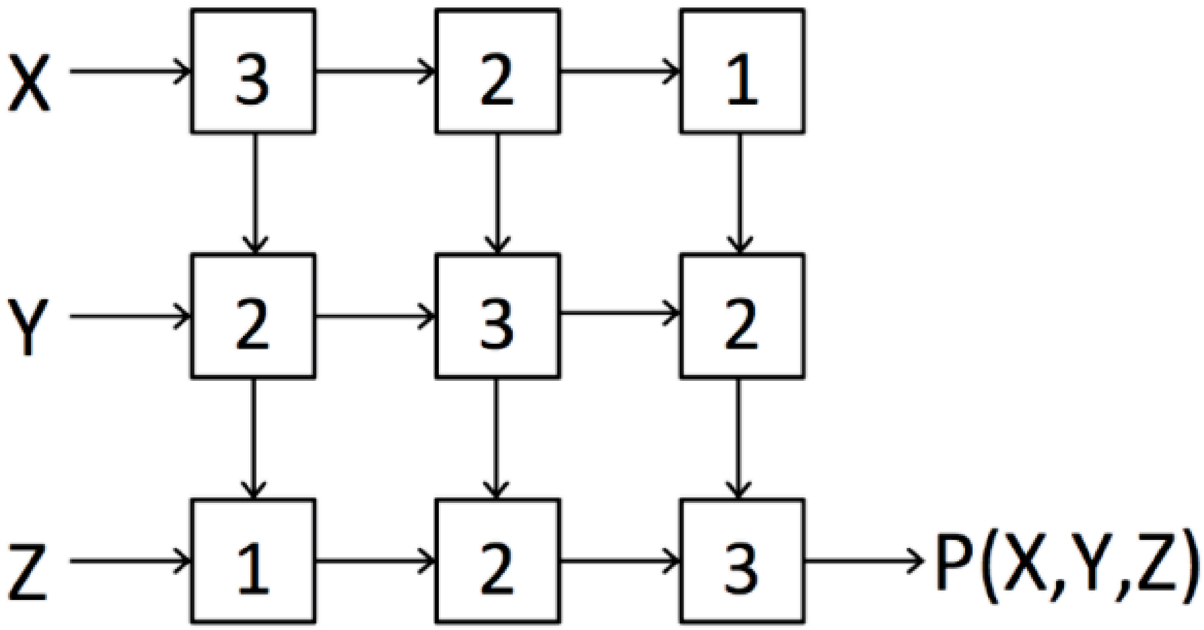
Answers are displayed within the problem

LE7.2.3 Building a Pipeline

3/5 points (ungraded)

The top-secret diagram of the NSA's phone call tracking circuitry is shown below. We don't know what the boxes do, but we do know their t_{PD} (annotated inside each component) and how they are connected.

You've been hired to do a design study looking into pipelining the circuit above. Assume that the pipeline registers have $t_{CD} = 0$, $t_{PD} = 0$, $t_{SETUP} = 1$, $t_{HOLD} = 0$. Recall that using our pipelining convention, each pipelined circuit must have at least 1 register on the P(X,Y,Z) signal.



What are the latency and throughput of a 1-pipeline implementation?

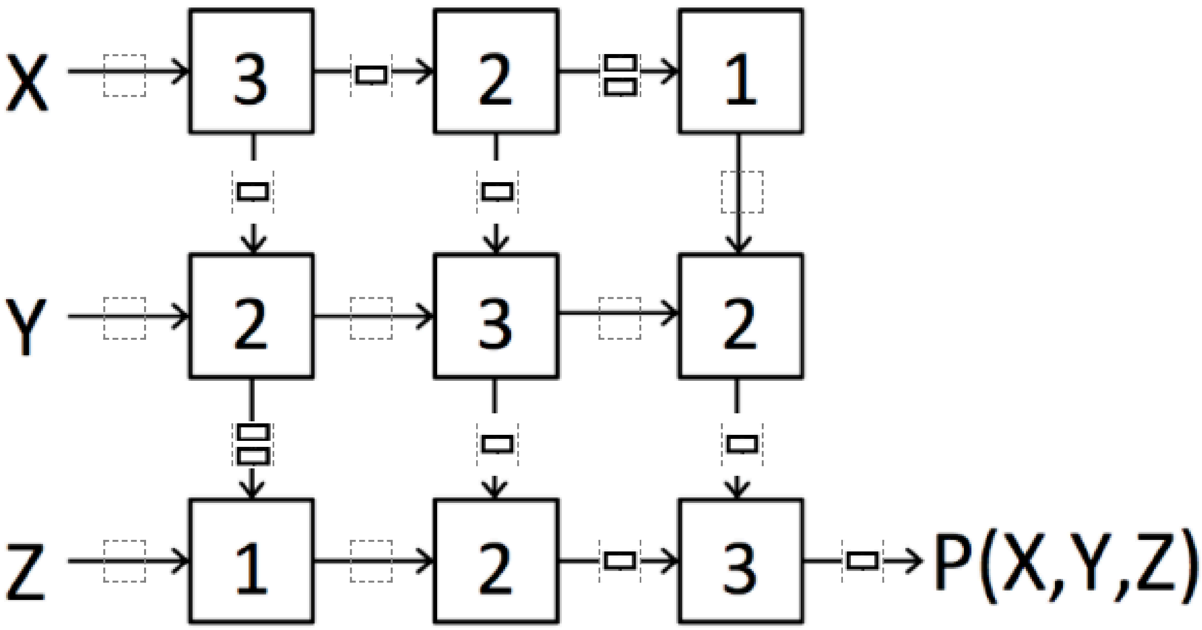
Latency: 14 Answer: 14

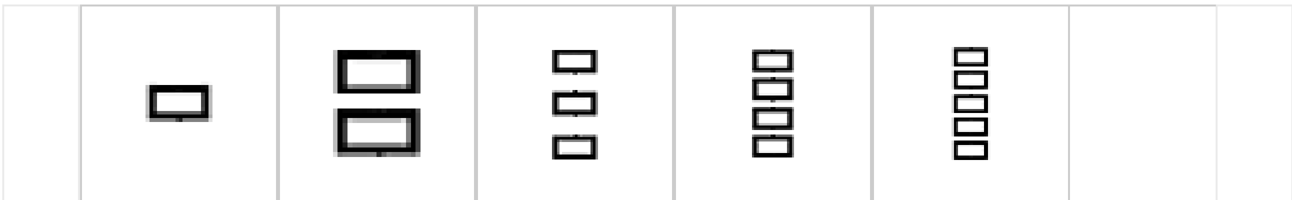
Explanation
The latency of a pipelined circuit is equal to the number of pipeline stages times the clock period. In this problem, we are designing a 1-pipeline implementation that has a single pipeline stage with a pipeline register at the output. The clock period for this pipelined circuit must be long enough to cover the propagation delay of any upstream pipeline registers (if the inputs are coming from another pipelined circuit), plus the propagation delay of the longest combinational path, plus the setup time of the output pipeline register. The longest combinational propagation delay is from input X to the output P(X, Y, Z) and passes through the three 3 modules and two 2 modules, so the combinational propagation delay is $3\,(3) + 2\,(2) = 13$. This means that the clock period is $0 + 13 + 1 = 14$. So the latency of a 1-pipeline implementation is $1 * 14 = 14$.

Throughput: 1/14 Answer: 1/14

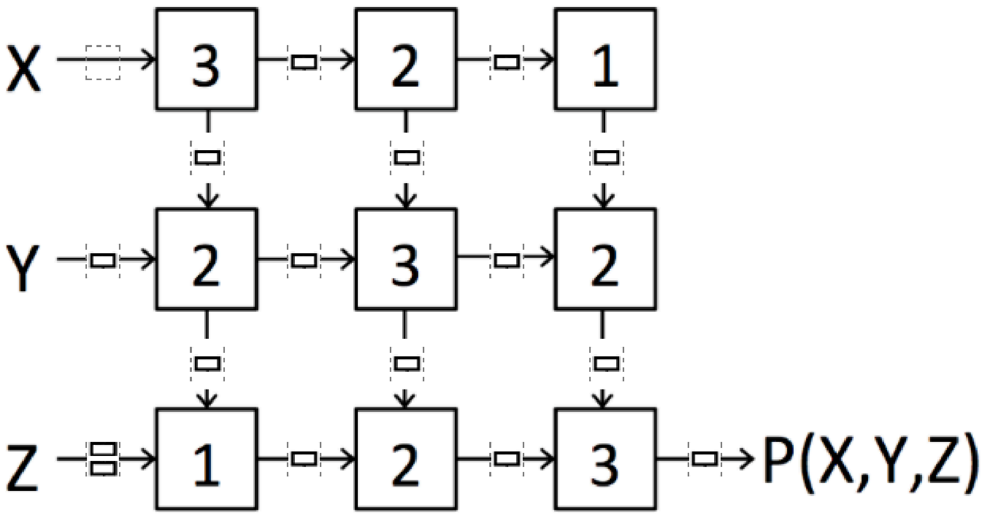
Explanation
The throughput of a pipeline circuit is 1/(clock period). We just determined that our clock period is 14, so the throughput is 1/14.

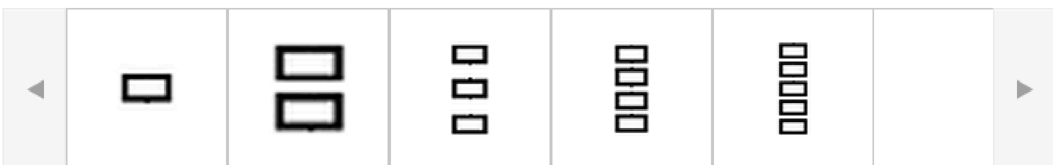
Add pipeline boundaries to the circuit, choosing the boundaries to achieve maximum throughput. Use the minimum number of registers necessary to achieve this throughput. Provide your answer by dragging the correct number of pipeline registers onto each of the dashed square boxes.





Explanation
One possible solution to pipelining the circuit is shown here.





Note that in order to pipeline the circuit for maximum throughput, each pipeline stage can at most be 3 units long. This means that the top left 3 unit and the bottom right 3 unit must end up in their own pipeline stage. The middle 3 unit can be together with other shorter units as long as those can be executed in parallel with this 3 unit. This is true of the top right 1 module and the bottom left 1 module. Either or both of these can be placed into the same pipeline stage as the middle 3 unit.
Note that from every input to output path there must be exactly the same number of pipeline registers for your solution to be valid.
What are the latency and throughput of your pipelined circuit?

Latency: 16 ✖ Answer: 20

Throughput: 1/4 ✔ Answer: 1/4

Explanation
When pipelining for maximum throughput, you want to make the combinational propagation delay between any two pipeline registers be less than or equal to the propagation delay of your slowest component, which is the 3 component in this case.
This means that the clock period of your pipelined circuit will be
$= t_{PD,reg} + t_{PD,combinational} + t_{setup,register} = 0 + 3 + 1 = 4$. The latency of the circuit is equal to the number of pipeline stages times the clock period. The number of pipeline stages you end up with when pipelining this circuit is 5, so the latency is
$= 5\,(4) = 20$.
The throughput is 1/(clock period) = 1/4.

Submit

Answers are displayed within the problem

Discussion

Hide Discussion

Topic: 7. Performance Measures / LE7.2

Add a Post

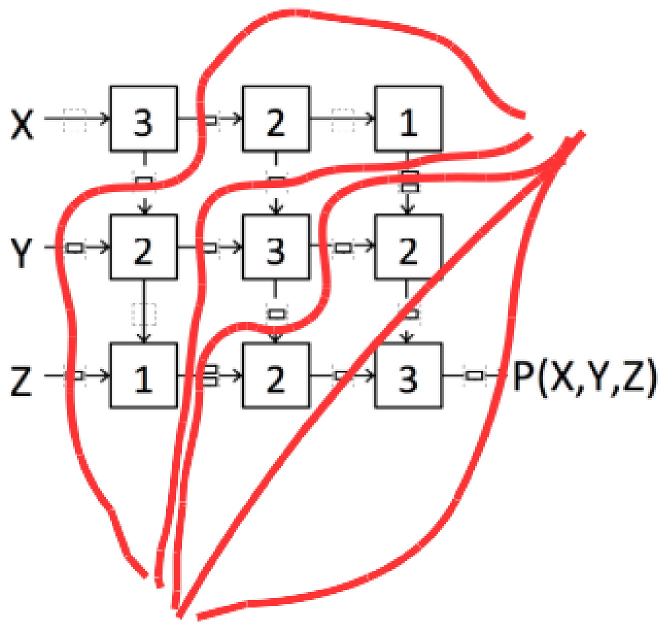
< All Posts

LE7.2.3 Pipelining

question posted 7 years ago by **jandevries**

** spoiler warning** This post contains a picture of the possible answer to the question, you may want to not read further if you want to figure it out on your own first.

The answer shown has 16 registers, but it seems there is also a possibility to pipeline the circuit with 15 registers, obtaining the same latency and throughput.



As the question wants us to use the minimum number of registers, and the give answer uses 16, maybe I missed something. Did I make a mistake? (it is graded as correct but that can be a mistake too)

This post is visible to everyone.

Grimeson
7 years ago - marked as answer 7 years ago by **silvinahw** (Staff)

No, you didn't make a mistake. The only difference between your scheme and the official solution is the bottom-left 3 nodes. And if you inspect these paths, you'll see that the number of register delays match between the two solutions. Also, none of your paths has a combinational latency that exceeds 3. Therefore, your solution is correct. Good find!

Small note though: I'm not sure if posting a solution here is the best idea since people who want to solve it themselves may inadvertently see your solution which will spoil the fun.

Show Comments (3)

Preview

Submit