We are given this circuit which consists of nine combinational modules connected as shown.

The number in each block corresponds to the propagation delay (in microseconds) of that block.

The first question we want to ask ourselves is what are the latency and throughput of this combinational circuit?

The longest path through this circuit passes through two 3 microsecond modules, one 2 microsecond module, and two 1 microsecond modules.

Therefore the latency = 2(3) + 1(2) + 2(1) = 10 microseconds.

The throughput is 1/Latency = 1/(10 microseconds).

Now we want to pipeline this circuit for maximum throughput.

Recall that the clock period must allow enough time for the propagation delay of the pipeline register, plus the propagation delay of any combinational logic between the pipeline registers, plus the setup time of the pipeline register.

Since our pipeline registers are ideal, the propagation delay and setup time of the pipeline registers is 0, so our clock period will be equal to the longest combinational logic delay between any pair of pipeline registers.

To minimize the clock period, we want to add pipeline contours so that each pipeline stage can be clocked at the rate of our slowest component which is 3 microseconds.

This means that within a single pipeline stage all combinational paths must have at most a propagation delay of 3 microseconds.

Recall, that when pipelining a circuit, all outputs must have a pipeline register on them, so our first contour is drawn so as to cross the single output C(X).

Each pipeline stage should have at most a latency of 3 microseconds.

This means that the bottom right 1 microsecond module and the 3 microsecond module above it must be in separate pipeline stages so our next contour crosses between them.

We then complete that contour by joining each end of the contour to one of the two ends of our original output contour.

Every time a wire is crossed, that indicates that we are adding a pipeline register.

There is more than one way of doing this because for example, the bottom three 1 units could all be in the same pipeline stage, or just the two rightmost bottom 1 unit components, or just the single bottom right 1 unit component.

Let's try to pipeline this circuit in two different ways to make sure that we end up with the same latency and throughput results.

For our first implementation, let's put the bottom right 1 unit in its own pipeline stage.

Next we have a pipeline stage that includes our second row 3 unit.

We can include the middle bottom 1 unit in this same pipeline stage because those two units are independent of each other and can be executed in parallel.

Now we see that we have a bunch of 2 and 1 unit components.

We want to isolate these from the top left 3 unit component, so we draw our final contour to isolate that 3 unit.

Note that at this point all the remaining 2 and 1 unit modules add up to at most 3 microseconds along any path in the second pipeline stage.

This means that we are done and we do not need to add any further pipeline stages.

So our pipelined circuit ended up with 4 pipeline stages and our clock period = 3 microseconds.

This means that our pipelined latency is 4 * T = 4 * 3 = 12 microseconds.

Our throughput for the pipelined circuit is 1/T = 1/(3 microseconds).

Note that the latency of the pipelined circuit is actually slower than the combinational latency.

This occurs because our pipelined implementation has some unused cycles in the last pipeline stage.

However, our throughput is significantly better as a result of being able to now have our clock period equal to the length of the slowest component.

Recall that we said that this was not the only way that we could draw our contours.

Let's try an alternate solution.

For our second solution, let's try to merge the bottom three 1 units into one pipeline stage.

Next we need to isolate our middle 3.

The remaining 2 and 1 unit components can all be merged into another pipeline stage.

Finally, the top left 3 ends up in its own pipeline stage.

As before we end up with 4 pipeline stages that can be clocked with a period of T = 3 microseconds.

This demonstrates that both component divisions end up with a latency of 12 microseconds and a throughput of 1/(3 microseconds).

Now suppose you found pipelined replacements for the components marked 3 and 2 that had 3 and 2 stages, respectively, and could be clocked at a 1 microsecond period.

Using these replacements and pipelining for maximum throughput, what is the best achievable performance?

With these new components that can each be clocked with a period of 1 microsecond, our clock period T goes down to 1 microsecond.

The number of stages in our pipeline, however, now rises to 10 stages because these new components each have multiple pipeline stages in them.

So our new latency is L = 10 * T = 10(1) = 10 microseconds.

The throughput is 1/T = 1/(1 microsecond).