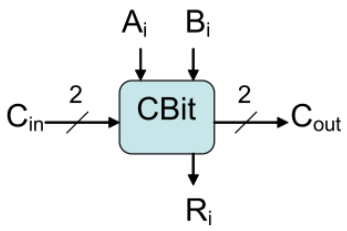
WE7.1

Bookmark this page

Video explanation of solution is provided below the problem.

Pipelining

12/12 points (ungraded)

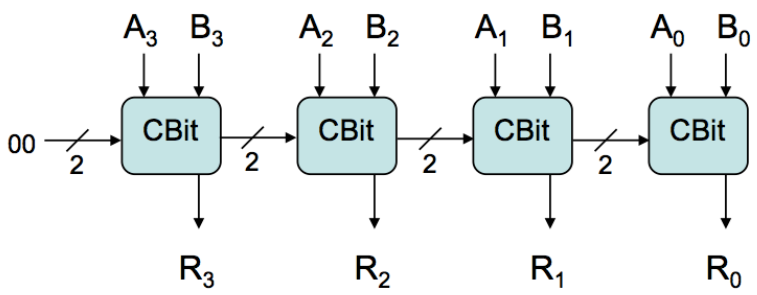


MaxOut is a Cambridge startup whose products are binary comparators which determine the largest of several unsigned binary integers. A building block common to all MaxOut products is the combinational CBit module depicted to the left.

Each CBit module takes corresponding bits of two unsigned binary numbers, A and B, along with two C_{in} bits from higher-order CBit modules. Its output bit, R, is the appropriate bit of the larger among A and B, as determined from these inputs; it passes two C_{out} bits to lower-order CBit modules.

The propagation delay of each CBit module is 4 ns. The two C_{out} bits indicate, respectively, if A>B or B>A in the bits considered thus far.

The first MaxOut product is MAXC, a combinational device which determines the maximum of its two 4-bit unsigned binary inputs. It is constructed using 4 CBit modules:



In the above diagram, unused inputs are tied to 0. The output $R_{3:0}$ is the larger of $A_{3:0}$ or $B_{3:0}$.

(A) What propagation delay specification is appropriate for the combinational MAXC module? What is its throughput?

MAXC propagation delay spec (ns): 16 ✔ Answer: 16

MAXC throughput (provide your answer in the form 1/X) (1/ns): 1/16 ✔ Answer: 1/16

Explanation
Since the propagation delay of each CBit is 4ns. The propagation delay of the combinational MAXC which is a 4 CBit module is 4 times the propagation delay of a single CBit, so the propagation delay of MAXC is 16 ns. In combinational circuits, the throughput of a circuit is 1/Latency. So the throughout is 1/(16 ns).

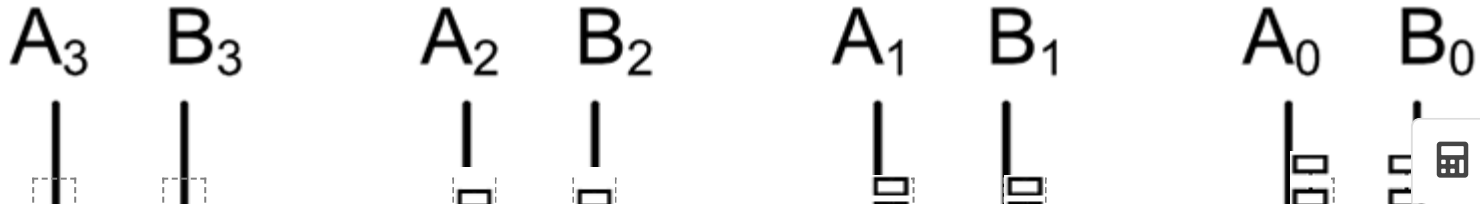
(B) If $A_{3:0}$ and $B_{3:0}$ are identical numbers, what two bits would you expect to see coming out of the (unused) C_{out} outputs from the low-order CBit module?

Low-order Cout bits for A=B: 0 ✔ Answer: 0 and 0 ✔ Answer: 0

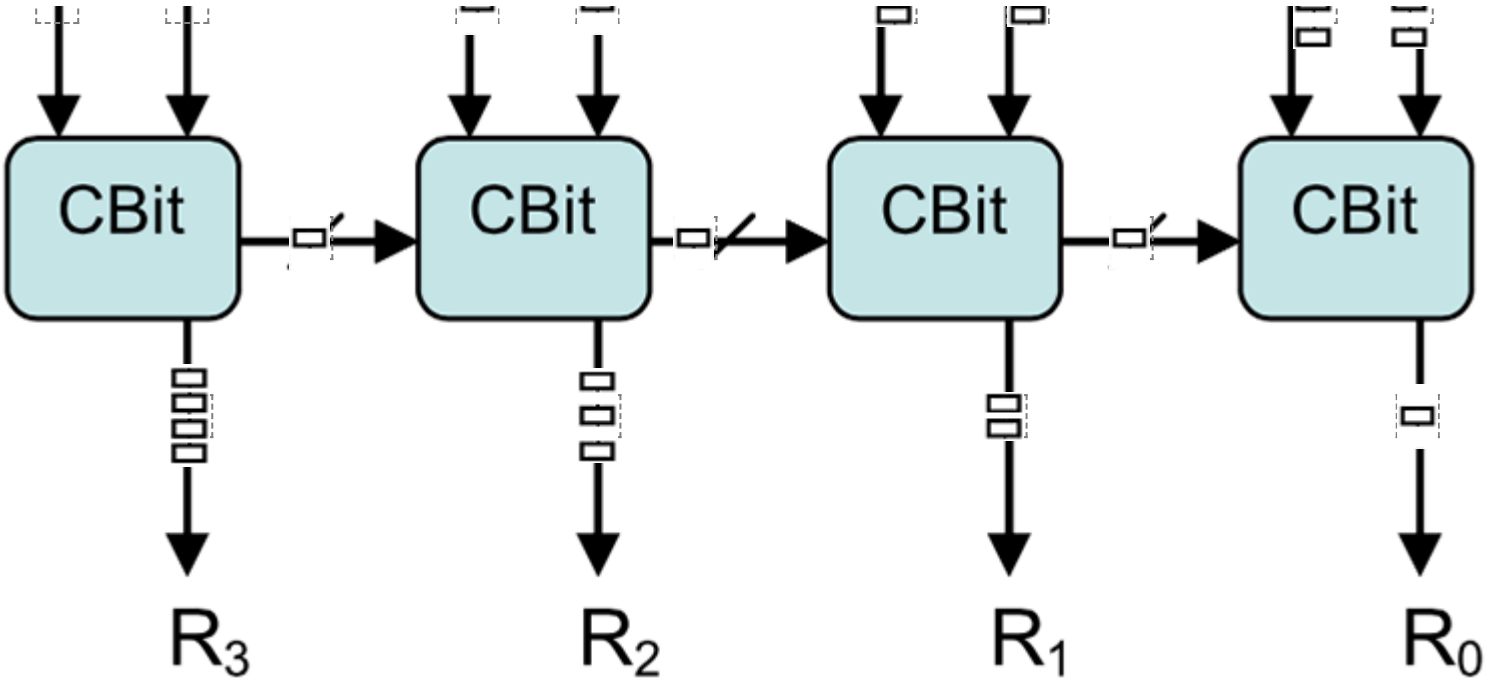
Explanation
If two numbers are identical, then neither A>B nor B>A are true so both C_{out} bits are 0.

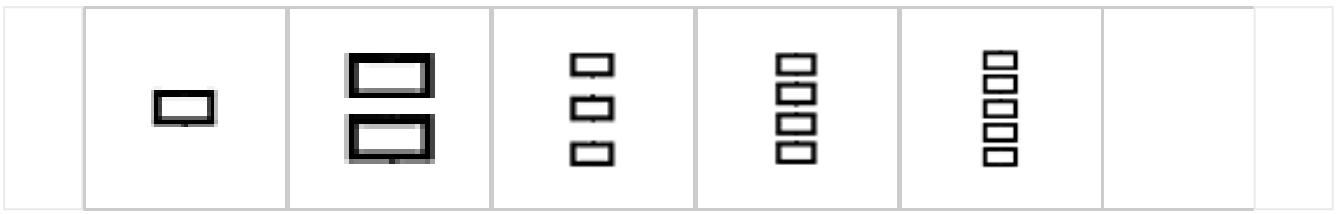
MaxOut's second product, MAXP, is identical to MAXC except that it includes the minimum number of registers necessary inserted to pipeline the circuit for maximum throughput.

(C) On the diagram below pipeline the MAXC for maximum throughput. Drag the correct number of ideal (zero delay, zero setup/hold time) registers onto each of the dashed square boxes so that you produce a valid pipeline with maximum throughput. Be sure to include at least one register on each output.



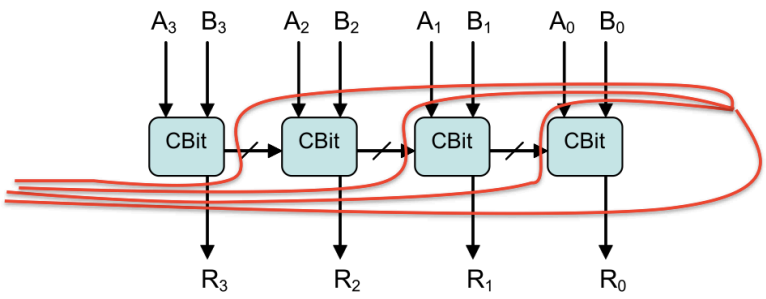
Calculator





✔

Explanation
In order to pipeline the MAXC for maximum throughput, we want to add pipeline registers to the circuit that isolate each individual CBit to be in its own pipeline stage. In order to make it a proper pipeline, we need to ensure that all paths from input to output have the same number of registers. Recall, that when pipelining a circuit, we want to add pipeline registers to all the outputs, so we begin by adding a contour that goes across all 4 outputs. We then work our way back adding contours representing pipeline registers that isolate each CBit while ensuring that the total number or registers from input to output remains the same along all paths. The resulting pipelined MaxC is shown here.



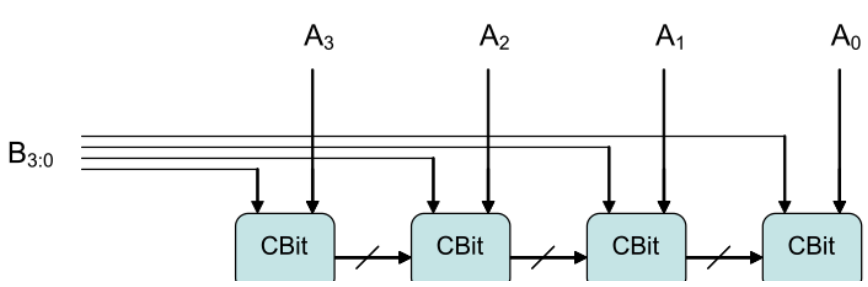
(D) What are the latency and throughput of your pipelined MAXC?

Pipelined MAXC latency (ns): 16 ✔ Answer: 16

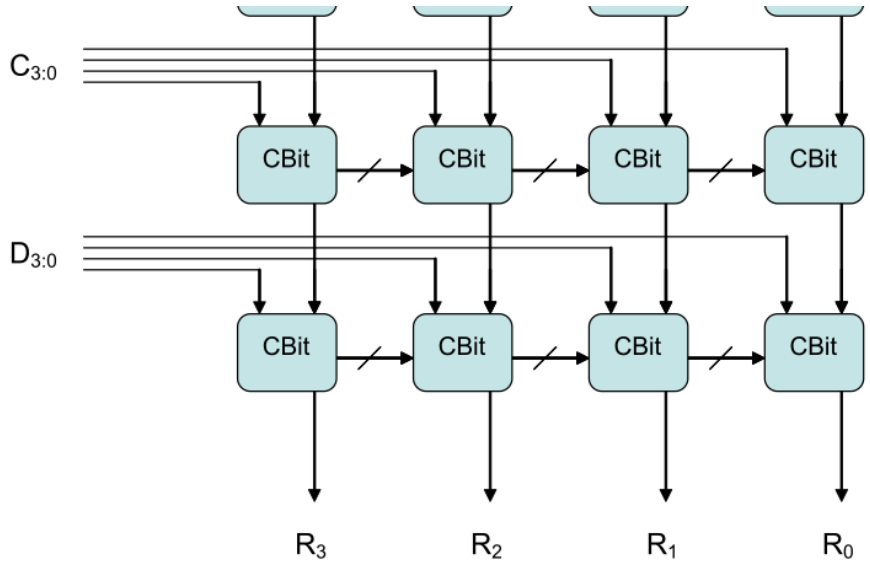
Pipelined MAXC throughput (provide your answer in the form 1/X) (1/ns): 1/4 ✔ Answer: 1/4

Explanation
Since the pipelined version of the MaxC has at most 1 CBit module in each pipeline stage, and the pipeline registers used are ideal (0 setup and hold time), it can be clocked with a period equal to the propagation delay of a single CBit module, so period = 4ns. The latency of a pipelined circuit is equal to the clock period times the number of pipeline stages. The MAXC has 4 pipeline stages, so its latency is 4 * 4ns = 16 ns.
The throughput of a pipelined circuit is 1/(clock period ns) = 1/(4 ns).

Expanding their product line, MaxOut's next product – the MAX4X4 -- is a combinational circuit capable of determining the maximum of four 4-bit binary inputs.



Calculator



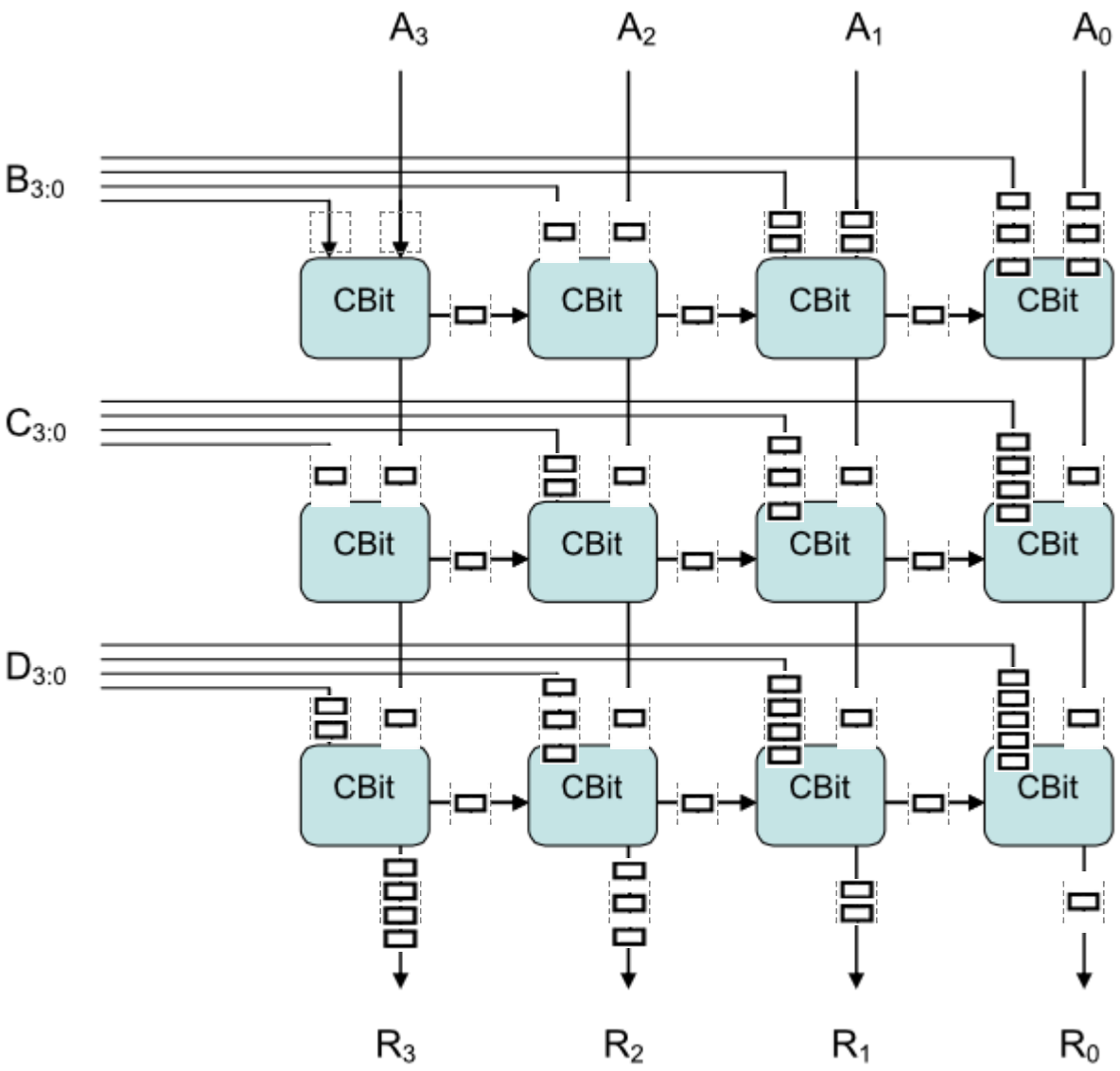
(E) What are the best latency and throughput that can be achieved using the combinational MAX4X4?

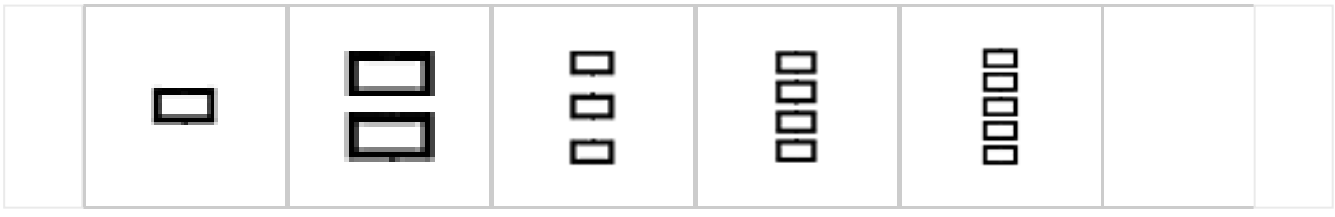
Latency (ns): 24 ✔ Answer: 24

Throughput (provide your answer in the form 1/X) (1/ns): 1/24 ✔ Answer: 1/24

Explanation
The propagation delay of the combinational MAX4X4 which is a 4×4 CBit module is the propagation delay along the longest path. All paths through the MAX4X4 pass through 6 CBit modules. Since the propagation delay is each CBit module is 4 ns, the propagation delay of MAX4X4 is 6 * 4 ns = 24 ns.
In combinational circuits, the throughput of a circuit is 1/Latency. So the throughout is 1/(24 ns).

(F) On the diagram below, pipeline the MAX4X4 for maximum throughput. Do this by dragging the correct number of ideal registers onto each of the dashed square boxes so that you produce a valid pipeline with maximum throughput. Then give the best latency and throughput that can be achieved by pipelining the MAX4X4.





✔

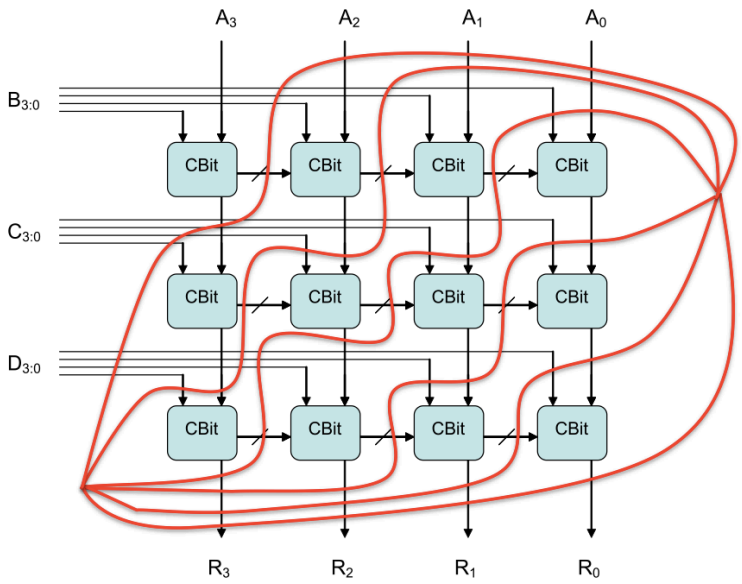
Calculator

Latency (ns): 24 ✔ Answer: 24

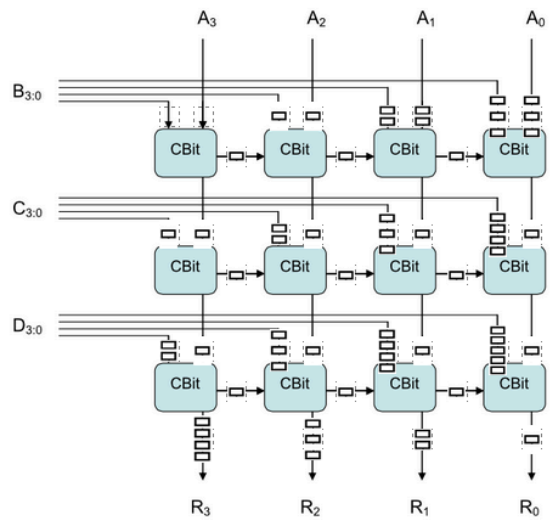
Throughput (1/ns): 1/4 ✔ Answer: 1/4

Explanation

In order to pipeline the MAX4X4 for maximum throughput, we want to add pipeline registers to the circuit that isolate each individual CBit to be in its own pipeline state. In order to make it a proper pipeline, we need to ensure that all paths from input to output have the same number of registers. Recall, that when pipelining a circuit, we want to add pipeline registers to all the outputs, so we begin by adding a contour that goes across all 4 outputs. We then work our way back adding contours representing pipeline registers that split up the CBit modules so that on any single path from input to output, only one CBit module is crossed per pipeline stage, while ensuring that the total number or registers from input to output remains the same along all paths. The resulting pipelined Max4X4 is shown here.



The same solution, illustrated with registers rather than contours, is shown here.



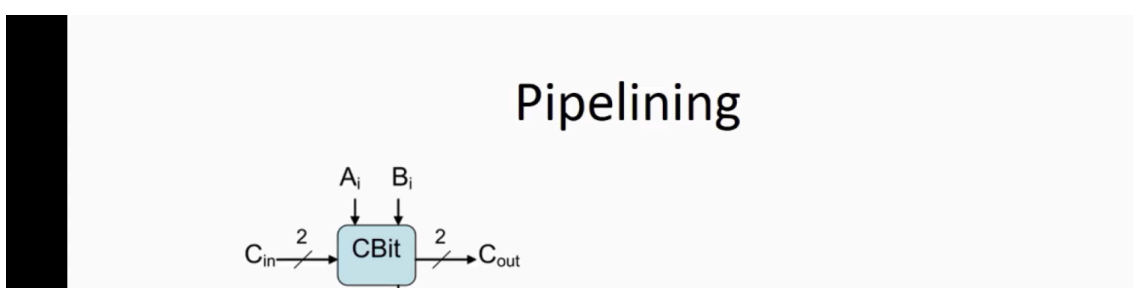
✔

Since each pipeline stage of the pipelined MAX4X4 has exactly 1 CBit module in it, and ideal pipeline registers (0 setup and hold time) are being used, this pipelined circuit can be clocked with a period = 4 ns. The latency of a pipelined circuit is equal to the clock period times the number of pipeline stages. From the diagram, we see that the pipelined MAX4X4 has 6 pipeline stages, so the latency is 6 ∗ 4 ns = 24 ns. The throughput of a pipelined circuit is 1/(clock period ns), 1/(4 ns).

Submit

Answers are displayed within the problem

Pipelining



Calculator

0:06 / 8:25 1.0x

We are given a CBit module shown here.

This module takes corresponding bits of two unsigned binary numbers, A and B, along with two C_{in} bits from higher-order CBit modules.

Its output bit R, is the appropriate bit of the larger among A and B.

It also has two additional output bits C_{out} which indicate respectively if A>B or B>A in the bits considered so far.

The propagation delay of each Cbit module is 4ns.

Video

Download video file

Transcripts

Download SubRip (.srt) file

Download Text (.txt) file

Discussion

Topic: 7. Performance Measures / WE7.1

Hide Discussion

Add a Post

Show all posts by recent activity

There are no posts in this topic yet.

Previous Next