MITx 6.004.1x

Computation Structures 1: Digital Circuits

Help

Course Progress Dates Discussion Course Notes

Course / 7. Performance Measures / Lecture Videos (33:12)

< Previous ✓ ✓ Next >

LE7.3

Bookmark this page

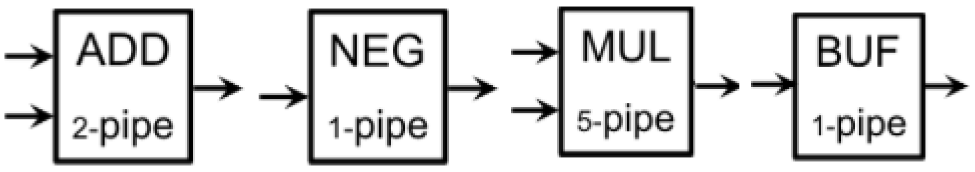
Calculator

LE7.3.1 Pipelined Components

6/6 points (ungraded)

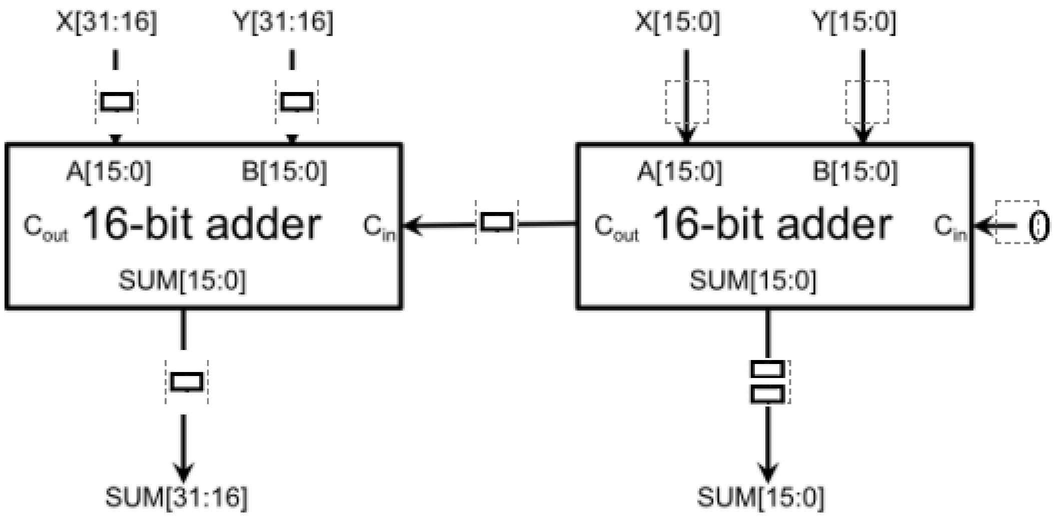
The Government's energy policy revolves around the efficient repetitive execution of a simple algorithm: for various values of A and B, it must compute the quantity Keystone(A,B) = (A+B)*(A-B). A consortium of entrepreneurs has proposed that a pipelined execution of this computation could solve the country's problems. You have been brought in as a consultant to evaluate their proposal. The Keystone proposal, as it is called, is that a set of pipelined modules be designed to perform basic operations, and these modules be assembled to perform the required computation of (A+B)*(A-B). It has been determined that the computation will use 32-bit twos-complement binary numbers as the representation of A, B, and the result of the computation.

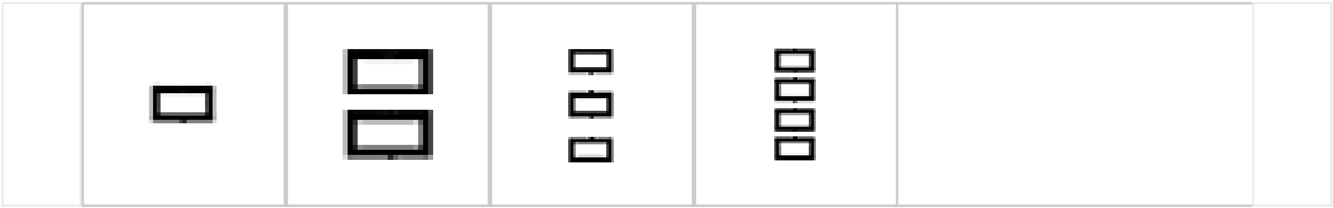
1. **ADD**: a 2-stage pipeline that adds two 32-bit quantities.
2. **NEG**: a 1-stage pipeline that negates a single 32-bit quantity.
3. **MUL**: a 5-stage pipeline that multiplies two 32-bit quantities, producing a 32-bit product.
4. **BUF**: a 1-stage pipeline that reproduces its 32-bit input one clock cycle later.



Each of the modules is a well-formed k-pipeline for its specified k, with k registers on each input/output path (including registers on each output). For example, Keystone proponents argue that the 2-stage ADD module can be made using two 16-bit adders by adding some additional registers:

Mark the locations of the minimal number of registers in the circuit below to make it a valid 2- stage pipelined ADD module while maximizing throughput. Provide your answer by dragging the correct number of pipeline registers onto each of the dashed square boxes. Be sure that each output has a register.

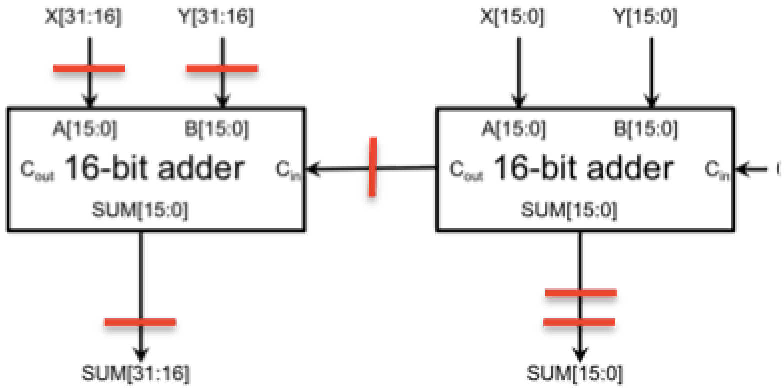




✓

Explanation

The image below shows the registers that need to be added in order to produce a 32-bit adder that has 2-pipeline stages.



During a vigorous Senate investigation, an engineer was forced to admit that the BUF module performed no actual computation, but contained a 32-bit register that served only to delay its input by one clock cycle. Senator Quagmire has demanded an explanation for why taxpayers should be asked to pay for developmen

Calculator

such a do-nothing module.

○ The extra registers are required to restore marginally-valid digital signals.

○ Additional registers are inserted to satisfy hold time requirements.

● Well-formed pipelines may require "do-nothing" pipeline stages.

○ You're absolutely right, Sir. We'll eliminate the BUF module immediately.

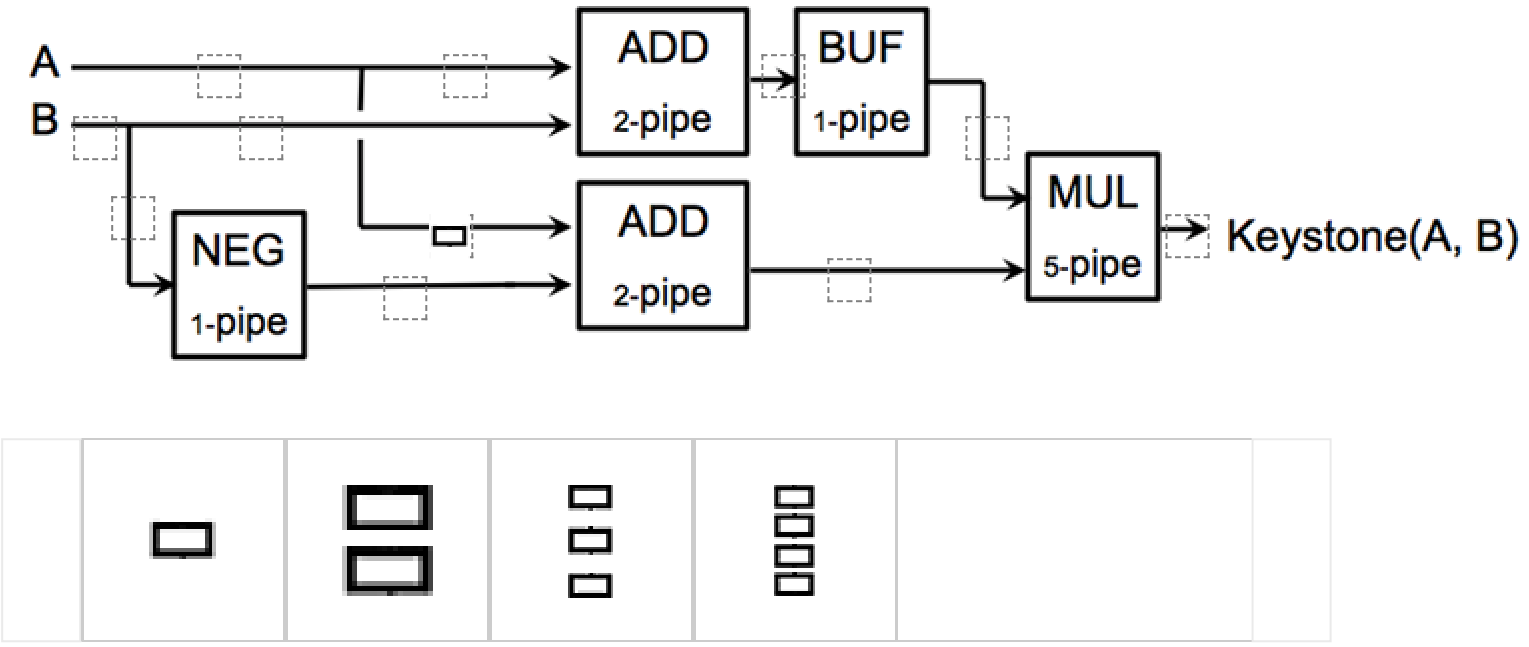
✓

Explanation
While other electronics circuits might insert buffers in order to restore valid digital signals, that is not the purpose of the buffer in this circuit. Similarly, we don't have any hold time requirements in this particular circuit. However, in order to sync inputs with each other, some signals need to be held in "do-nothing" stages while waiting for other parts of the circuit to propogate into a pipeline stage where the signals will combine synchronously. That means that we, as experts in the field, can NOT confirm the Senator's suspicion that the buffer is a wasteful element in the circuit.

Keystone proponents propose a specific implementation for the pipelined computation, using their module toolkit:

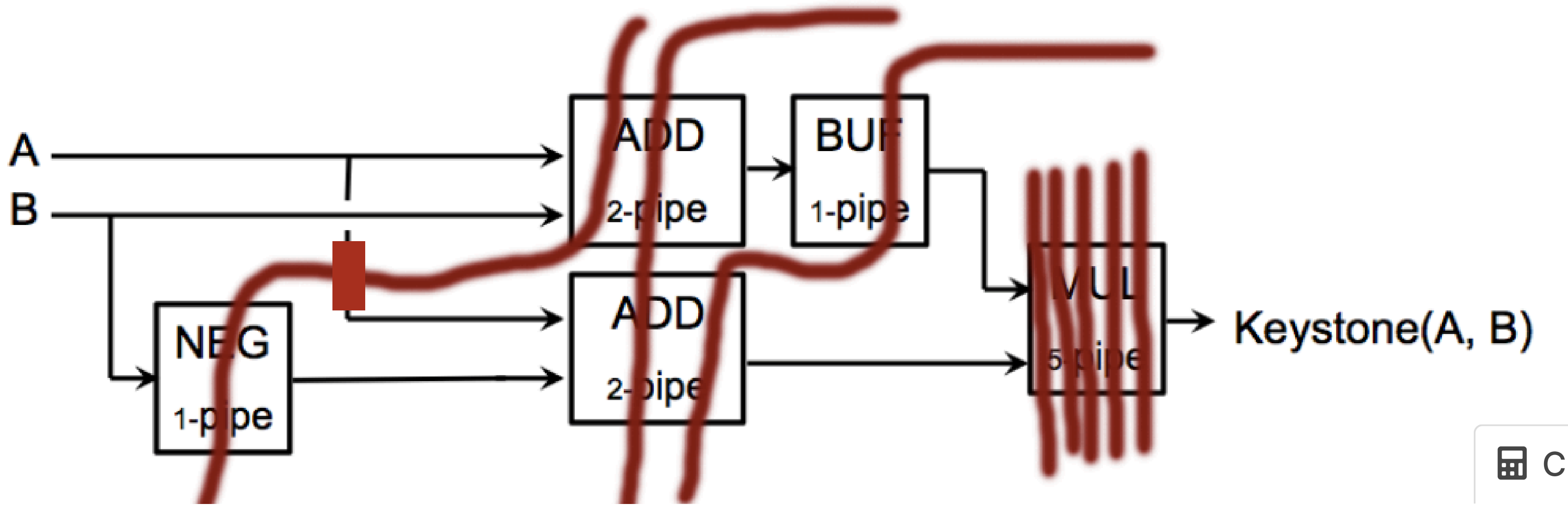
In a candid interview on Cheat the Press, one of the Keystone group admits that he is not 100% sure that provided schematic will work. His exact quote: "We tried to hire a 6.004x student, but they were all studying for some dumb exam!". Your chance to be a hero.

Is the diagram below a well-formed pipeline? If yes, do not add any wasteful buffers, and submit the diagram as-is. If not, show where to add a minimal number of BUF modules to make it well formed. For this problem treat each drag and drop register as a BUF module.



✓

Explanation
The tricky part about this pipelining problem is that registers for particular pipeline stages are already included in several of the submodules. If you draw pipeline stages with those existing pipeline stages in mind, you will observe that you only need a single additional register, as shown.



Calculator

Note that a register on the output is NOT needed in order to conform to our convention. The submodules are pipelined, and presumably contain a register at the output of the submodules because you always assume that conventions are followed. Therefore, there is a register at the output - it is contained within the submodule.

Assume that the clock period is 1 ns. What is the latency of the pipeline (with your fixes, if any)? What is the throughput?

Latency (ns): 8 ✓ Answer: 8

Explanation
There are eight pipeline stages, with 1ns per stage because that is the clock speed, giving a latency of 8 ns.

Throughput (1/ns): 1 ✓ Answer: 1

Explanation
The throughput of a pipelined circuit is always 1/(clock period) because new outputs are produced after each clock period, so the rate of receiving new outputs is 1/(clock period) which in this problem is 1/(1 ns).

The pipeline (with any amendments you suggested) is built at great cost, and works. But, at the last minute, it is learned that a the computation really needed is Keystone(XL(A), B) rather than Keystone(A,B): an extra XL stage must be added to the pipeline. The Government finds a combinational XL module whose t_{pd} is nearly 3 ns, but the XL module can't be pipelined. They insist that they need to maintain the pipeline's throughput, and again ask your advice on how to keep the current throughput while minimizing cost. How would you advise them to proceed?

○ Make a 1-stage pipelined XL by adding registers, and run with a 3ns clock period.

○ Construct a total of three Keystone(XL(A), B) pipelines, each with a 3ns clock period.

○ Remove all registers, and use a strictly combinational implementation.

● Use 3 combinational XL modules and some logic to emulate a 3-stage pipelined XL module, and add that to the existing pipeline.

○ Tell them that it's impossible to meet the updated specifications, and that they should have specified all of the circuit requirements before calling you in to provide a solution.

✓

Explanation
The taxpayers only need a throughput of 1ns. We would be able to accomplish that with three Keystone(XL(A),B) pipelines, but that is not the BEST solution. Instead, it is possible to use the pipelined circuits that were already built by adding the 3 XL modules to the input A and using logic to emulate a 3-stage pipelined XL module. As long as the outputs of each additional XL module reaches the original A input at a different time, the resulting circuit should have the same throughput as the original circuit.

Submit

ℹ Answers are displayed within the problem

Discussion

Hide Discussion

Topic: 7. Performance Measures / LE7.3

Add a Post

Show all posts ⌄ by recent activity ⌄

2 About First Question

Calculator

< Previous Next >

edX

About
Affiliates
edX for Business
Open edX
Careers
News

Legal

Terms of Service & Honor Code
Privacy Policy
Accessibility Policy
Trademark Policy
Sitemap
Cookie Policy
Your Privacy Choices

Connect

Idea Hub
Contact Us
Help Center
Security
Media Kit

GET IT ON Google play
Download on the App Store

Calculator