Report 2nd Week

1. Combinational circuit

1.1: Hazards in combinational logic

- In response to any single input change and for some combination of propagation delays, a circuit output may momentarily go to 0 when it should remain a const 1 => Static 1-hazard

Similarly with 0

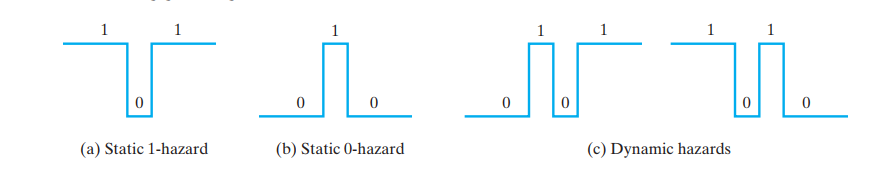


Figure 1: Types of Hazards

- Hazards can be detected using a Karnaugh map

1.2: Three state buffers

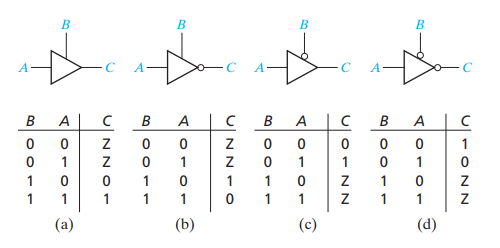
- 3state buffer may used to increase the driving capability of a gate output ( normally, a logic circuit will not operate correctly if the outputs of two or more gates or other logic devices are directly connected to each other).9

Figure 2: 4 kinds of three-state buffers

1.3: Read-Only Memories

A ROM basically consists of a decoder and a memory array. This decoder output line selects one of the words in the memory array, and the bit pattern stored in this word is transferred to the memory output lines.

1. Chaining and multicycling

If we have the total of any two functional units is shorter than the clock cycle => it’s possible to connect them in series ( This technique is called **chaining** )

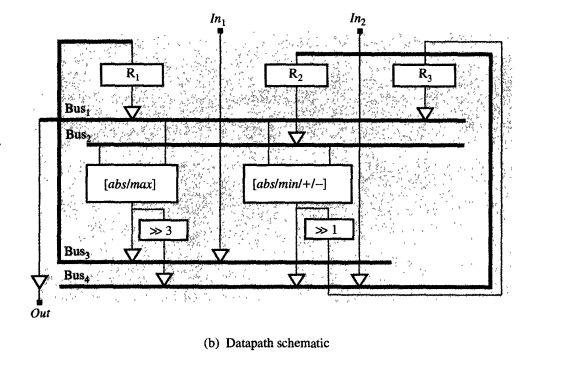


Figure 3: Example when applying chaining technique

In addition to chaining, which allows us to use faster units, we may want to use slower but less expensive units that take more than one clock cycle to generate their results => This technique is called **multicycling**, and these slower units called **multicycle units**.

1. Pipelining

3.1: Functional pipelining

We divide a functional unit into 2 or more stages, each separated by latches so that each stage can operate on a different set of operands

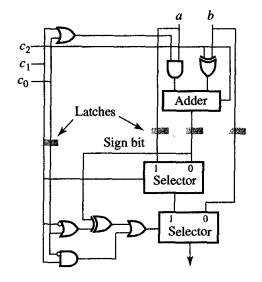


Figure 4: pipelined arithmetic unit

3.2 Datapath pipelining

We divide the entire ASM chart into several equal-size parts and then use a separate datapath stages to execute each part.

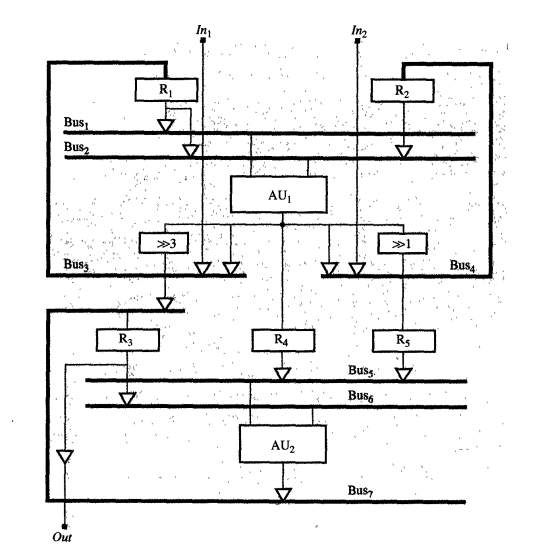


Figure 5: pipelined data-path

3.3 control-pipelining

Insert **status register** (signals go from the datapath to CU), **control register** (signals go CU and datapath) and **pipeline latches** (between the storage units and functional units)

1. Scheduling

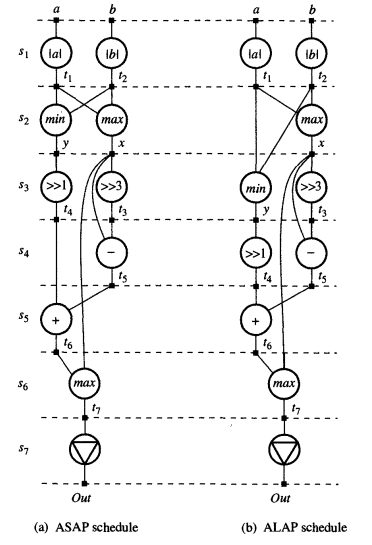


Figure 6: 2 algorithms to determine operation priority and range for scheduling

ASAP: \*definition

ALAP: \*definition

Mobility: M(op) = k – i (statesi  in ASAP, state sk in ALAP schedule )

One of the most popular algorithms for **RC scheduling** is **list-scheduling algorithm**

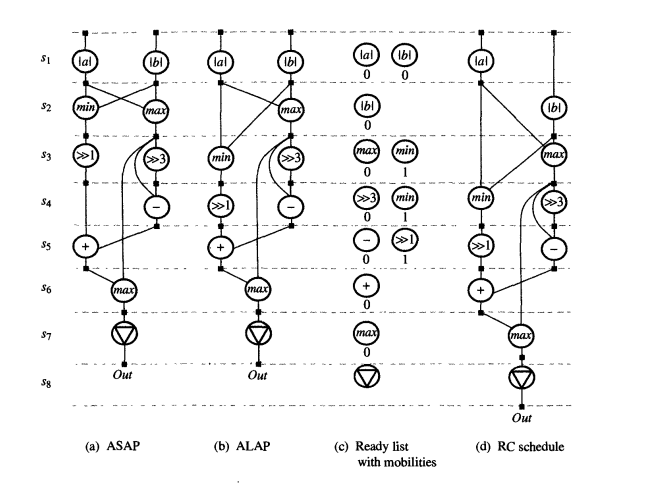


Figure 7: Example of Resource-constrained scheduling

To use TC scheduling: we first apply the ASAP and ALAP scheduling algorithms