Hardware Description Language

Pipeline Processor

EE18BTECH11025 & EE18BTECH11016

The basic concept of pipelining is to break up instruction execution activities into stages that can operate independently. Every instruction passes through the same stages much like an assembly line.

Pipelining attempts to keep every part of the processor busy with some instruction by dividing incoming instructions into a series of sequential steps performed by different processor units with different parts of instructions processed in parallel. It allows faster CPU throughput than would otherwise be possible at a given clock rate, but may increase latency due to the added overhead of the pipelining process itself.

One of the pipeline processor is RISC, it's fast speed is mostly because of pipelining.

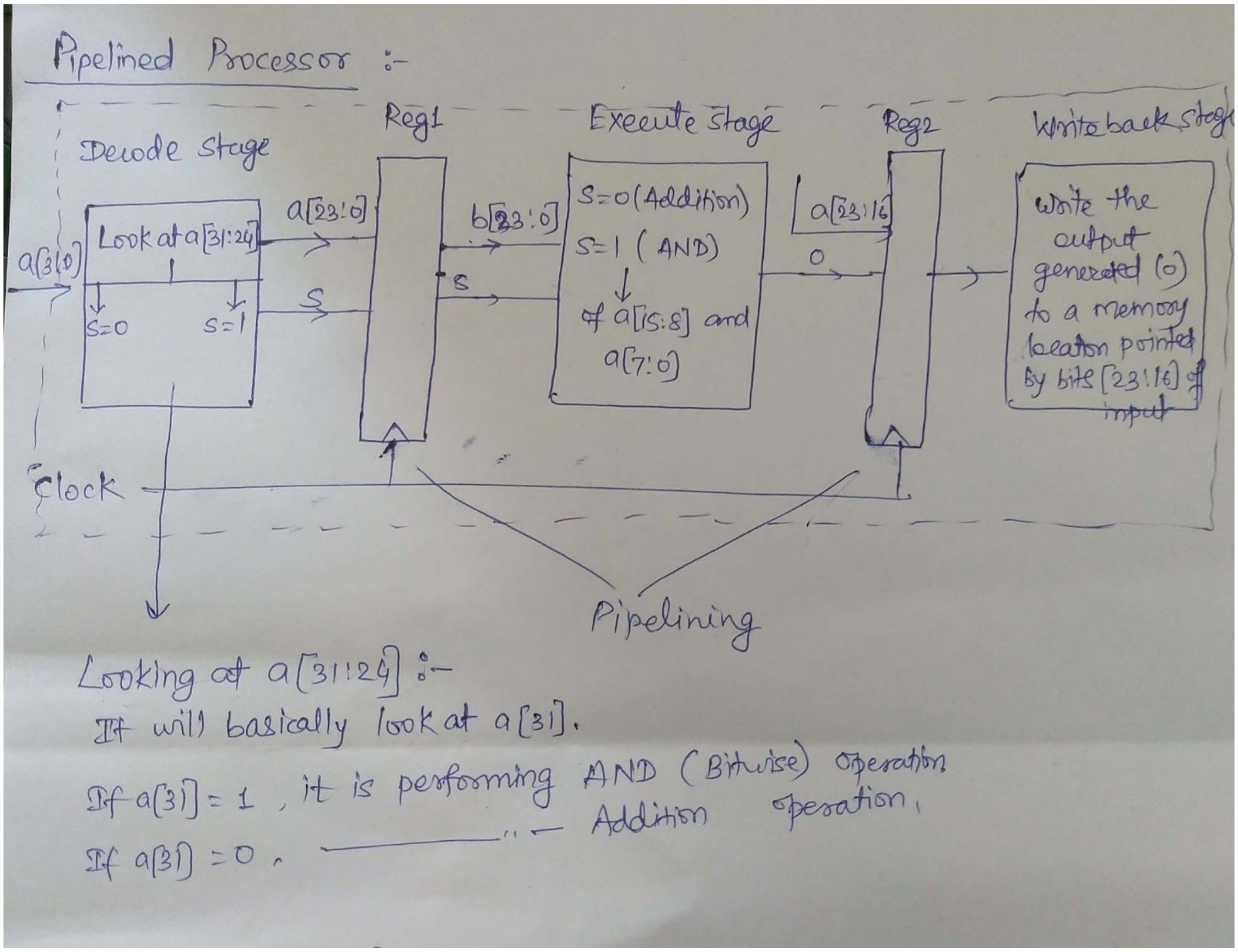
The classic RISC pipeline comprises:

1. Instruction fetch
2. Instruction decode and register fetch
3. Execute
4. Memory access
5. Register write back

The pipeline processor we have implemented has three stages:

1. Decoding the instruction signal and allocating memory for writeback
2. Execute stage (executes a function according to the decoded signal)
3. Write back stage (writes the output to the memory address)

It has two functions implemented, addition (arithmetic operation) and logical AND (bitwise, a logical operation), which has been done in the above said stages. We have also implemented a code that can read back the output that has been written to memory in the writeback stage.



Looking at a[31:24] :-
It will basically look at a[31].
If a[31] = 1 , it is performing AND (Bitwise) operation
If a[31] = 0 , ——————" — Addition operation.

Explanation of code in stages

1.Decode stage:

Basically we have to generate a control signal by looking at a [31:24]   (first 8 bits of the input which we are giving).
We are considering here that if a[31] = 1; then our control signal s = 1 and otherwise s = 0.
So that it will cover all the cases for a[31:24]

.
2.Execute Stage:
In the Execute stage, firstly the a[23:0] and s will go into register(Reg1). As soon as the positive edge of clock comes ,then the data at the input side of Reg1 will get loaded into second block where it will

perform the desired operation: Addition and Bitwise AND   (of a[15:8] and a[7:0])
if s=1; then the block will perform addition operation and gives its output to o.
if s=0; then the block will perform Bitwise AND operation and gives its output to o.

NOTE THAT THIS OPERATION IS PERFORMED AT THE POSITIVE EDGE OF THE CLOCK PULSE (pipeline concept).Because then it will not get inputs ::   a[23:0] and s.

3.WriteBack Stage:

In the WriteBack Stage, at the positive edge of the clock pulse,the Register (Reg2) passes a[23:16] and output o from Execute Stage to the writeback stage block.
The WriteBack will write the output generated (o) to a memory location pointed by bits[23:16] in input a.