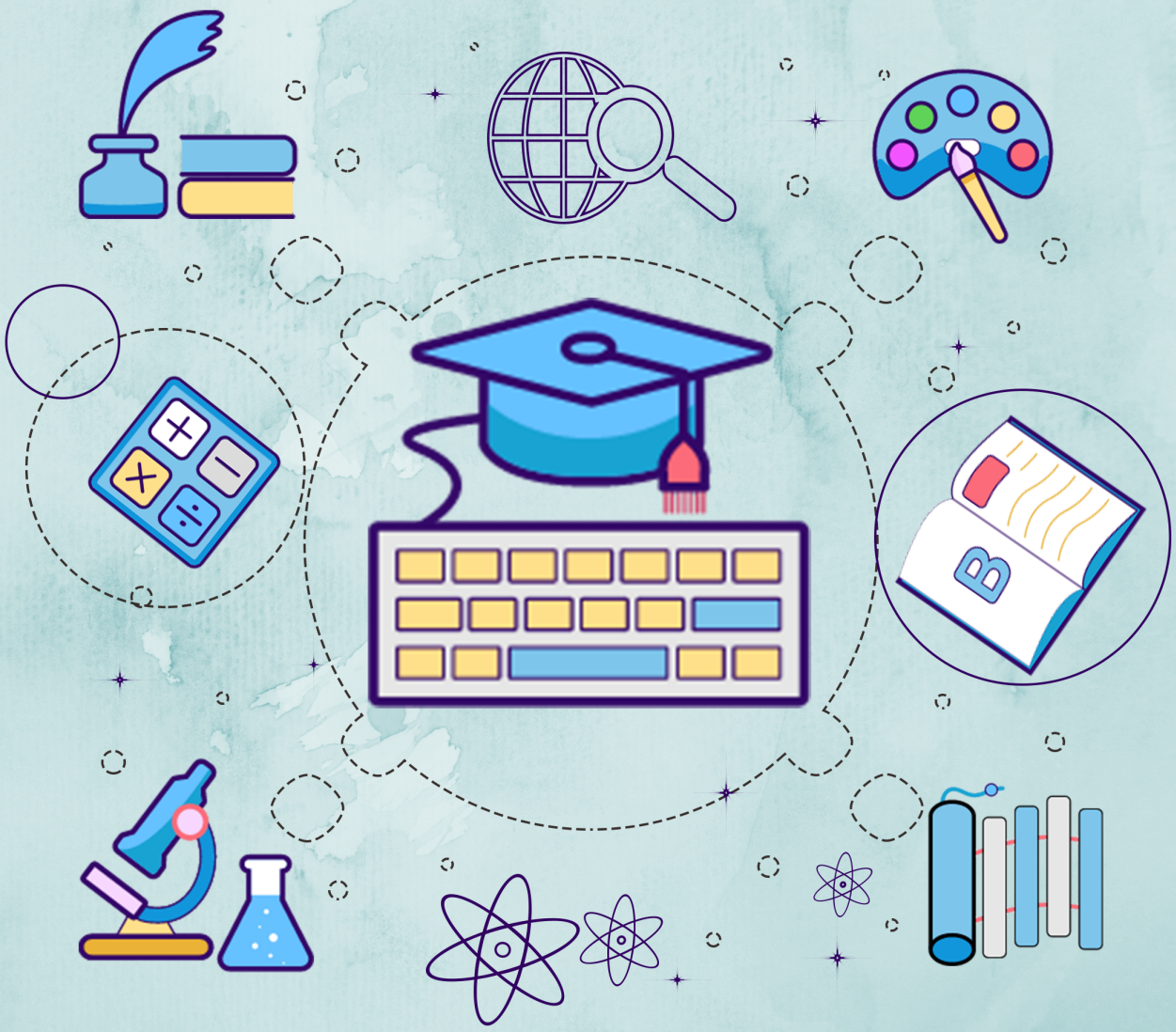
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

KeralaNotes



SYLLABUS | STUDY MATERIALS | TEXTBOOK PDF | SOLVED QUESTION PAPERS

www.keralanotes.com

KeralaNotes

KTU S4 CSE

COMPUTER ORGANIZATION AND ARCHITECTURE (CST 202)

QUESTION BANK

Related Link :

- KTU S4 CSE NOTES 2019 SCHEME
- KTU S4 SYLLABUS CSE COMPUTER SCIENCE
- KTU PREVIOUS QUESTION BANK S4 CSE SOLVED
- KTU CSE TEXTBOOKS S4 B.TECH PDF DOWNLOAD
- KTU S4 CSE NOTES | SYLLABUS | QBANK | TEXTBOOKS DOWNLOAD

www.keralanotes.com

MODULE 3

37. List the limitations of repeated addition method for multiplication.

a. Least sophisticated method

b. Just use adder over and over again

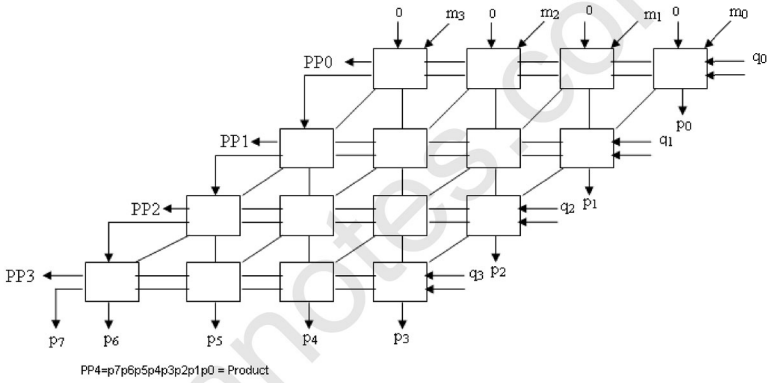
c. If the multiplier is n bits, can have as many as

d. 2n iterations of addition -- O(2n)

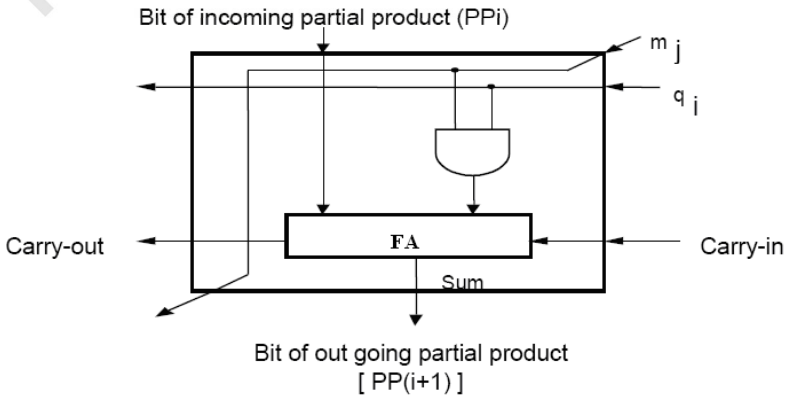
e. Not used in an ALU

38. Describe array processor with a neat diagram.

Binary multiplication of positive operands can be implemented in a combinational two dimensional array.

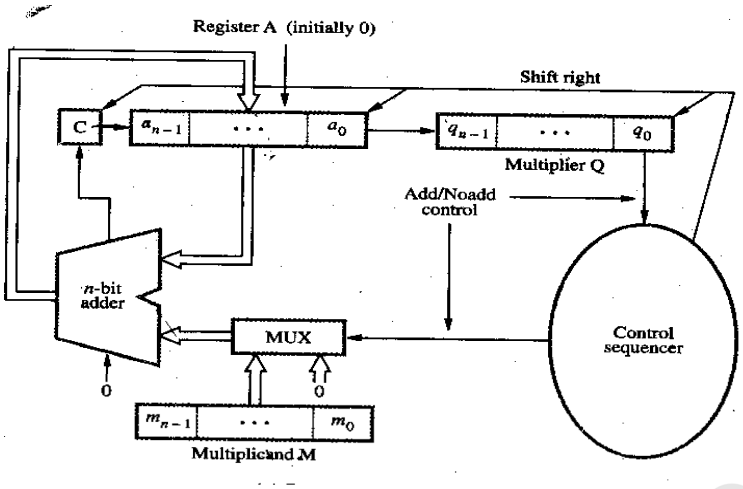


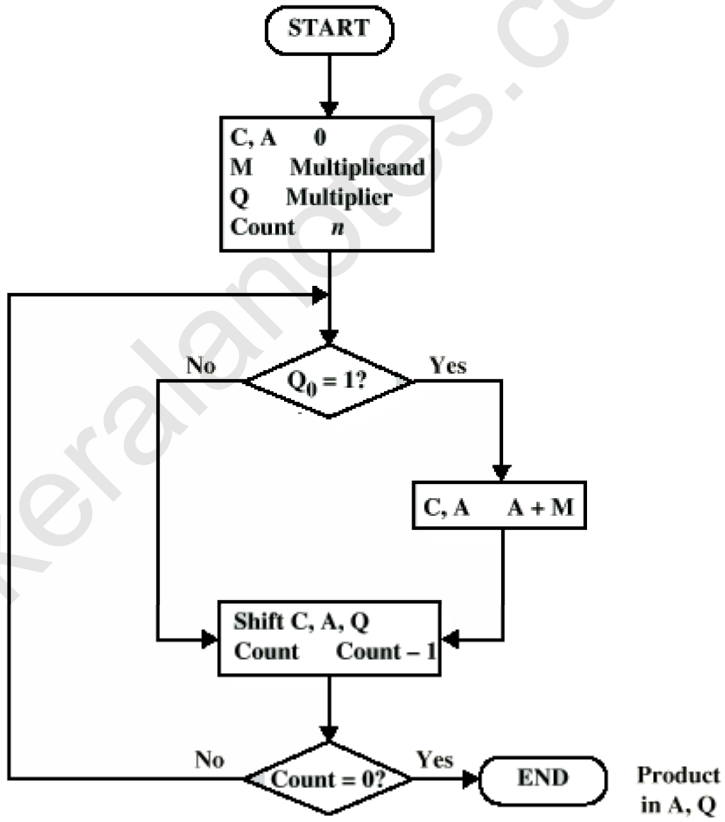
The AND gate in each cell determines whether a multiplicand bit **mj i**s added to the incoming partial product bit, based on the value of the multiplier bit **qj**. Each raw i adds the multiplicand to the incoming partial product **ppi** to generate pp(i+1) if qi=1. If qi=0 ppi is passed downward unchanged.



For More Study Materials :https://www.keralanotes.com/

39. Expalin sequential multiplier with an example.



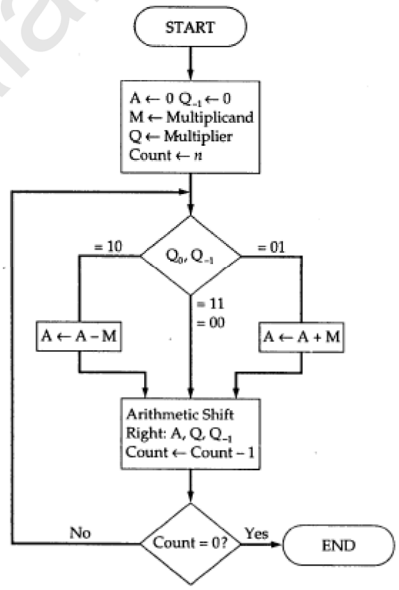


For More Study Materials :https://www.keralanotes.com/

C	A	Q	M		
0	0000	1101	1011	Initial Values	
0	1011	1101	1011	Add	First Cycle
0	0101	1110	1011	Shift	
0	0010	1111	1011	Shift	Second Cycle
0	1101	1111	1011	Add	Third Cycle
0	0110	1111	1011	Shift	
1	0001	1111	1011	Add	Fourth Cycle
0	1000	1111	1011	Shift	

40. Illustrate Booth algorithm with a suitable example.

When moving from 0 to 1 then -1 is selected and moving from 1 to 0 then +1 is selected as the multiplier scanning from left to right. This algorithm clearly extends to any number of blocks of 1s in a multiplier, including the situation in which a single 1 is considered as a block. If the first bit is 1 then consider the previous bit is 0.



The Multiplicand is placed in M register and multiplier is loaded into register Q. Registers A and Q_{-1} is cleared initially. A bit of multiplier is examined together with the bit in Q_{-1}. If

For More Study Materials :https://www.keralanotes.com/

these bits are same (0-0,1-1) then all bits of A,Q,Q_{-1} are shifted right 1 bit. If two bits are differ then the multiplicand is added to or subtracted from A depending on 0-1 or 1-0 then right shift occurs. In either case the right shift A_{n-1} to A_{n-2} occurs and A_{n-1} maintains the same bit for maintaining the sign. This is called Arithmetic shift.

A	Q	Q_{-1}	M		
0000	0011	0	0111	Initial Values	
1001	0011	0	0111	A A - M	First Cycle
1100	1001	1	0111	Shift	
1110	0100	1	0111	Shift	Second Cycle
0101	0100	1	0111	A A + M	Third Cycle
0010	1010	0	0111	Shift	
0001	0101	0	0111	Shift	Fourth Cycle

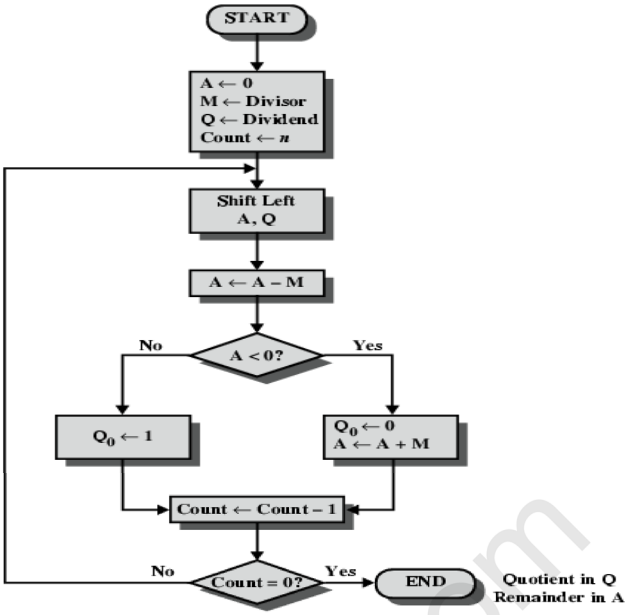
41. What are the advantages of Booth Algorithm

- It handles both positive and negative multiplier uniformly.
- it reduce the number of partial product,
- The speed gained by skipping 1's depends on the data.

42. Illustrate restoring division algorithm with an example.

Restoring Division: Register A is initially loaded with 0 and it consists of n+1 bits, where n is the number of bits in the dividend. Dividend is loaded in register Q and Register M is loaded with the divisor. After division is complete n bit quotient is in register Q and remainder is in register A. extra bit on A and M accommodates the sign bit during subtractions.

For More Study Materials :https://www.keralanotes.com/

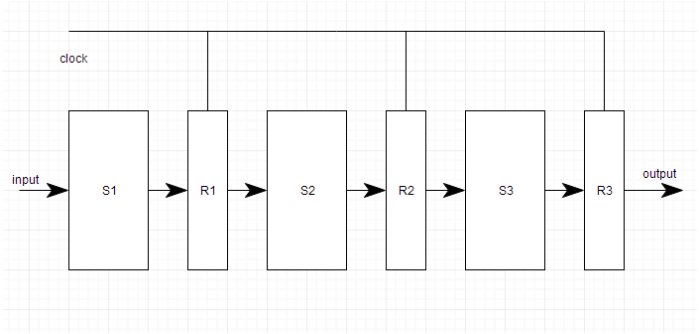


A	Q	M=00011
00000	1000	Initial value
00001	000-	Shift
11110	0000	Subtract/set q_0
00001		Restore
00010	000-	Shift
11111	0000	Subtract/set q_0
00010		Restore
00100	000-	Shift
00001	0001	Subtract/set q_0
00010	001-	Shift
11111	0010	Subtract/set q_0
00010		Restore

43. What is Pipelining?

Pipelining is a technique where multiple instructions are overlapped during execution. Pipeline is divided into stages and these stages are connected with one another to form a pipe like structure. Instructions enter from one end and exit from another end. In pipeline system, each segment consists of an input register followed by a combinational circuit. The register is used to hold data and combinational circuit performs operations on it. The output of combinational circuit is applied to the input register of the next segment.

For More Study Materials :https://www.keralanotes.com/



44. List the advantages of pipeline.

- Multiple tasks operating simultaneously using different resources
- Pipelining increases throughput

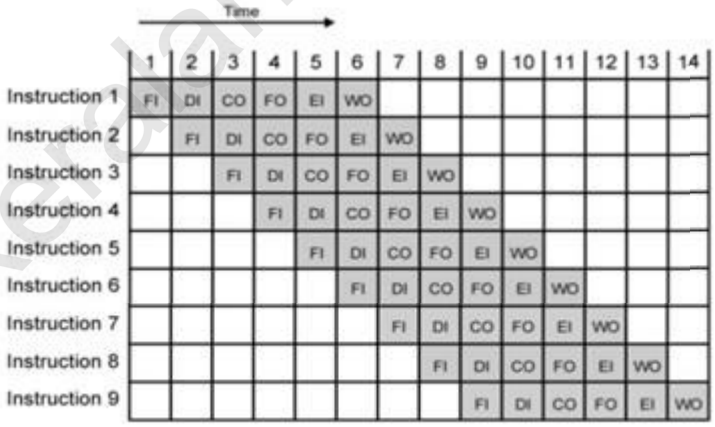
45. Different Types of Pipelines.

It is divided into 2 categories:

- Arithmetic Pipeline
- Instruction Pipeline

46. Explain instruction pipelining.

Instruction pipelining is a technique used in the design of modern microprocessors, microcontrollers and CPUs to increase their **instruction** throughput (the number of **instructions** that can be executed in a unit of time).



47. What are the various stages in instruction pipeline?

Stage 1 :**Instruction Fetch (FI)-**In this stage the CPU reads instructions from the address in the memory whose value is present in the program counter.

Stage 2 **:Instruction Decode (DI)-**In this stage, instruction is decoded and the register file is accessed to get the values from the registers used in the instruction.

For More Study Materials :https://www.keralanotes.com/

Stage 3: **Calculate Operands(CO)**-In this stage, effective address of the operands are calculated.

Stage 4: **Fetch Operands (FO)-**memory operands are read from memory

Stage 5 :**Instruction Execute (EI)-**In this stage, ALU operations are performed.

Stage 6: **Write Back (WO)-**In this stage, computed/fetched value is written back to the register present in the instructions.

48. How pipelining speeding up the performance of a computer?

Speedup (S) of the pipelined processor over non-pipelined processor, when 'n' tasks are executed on the same processor is:

S= Performance of pipelined processor/ Performance of non pipelined processor

Consider a 'k' segment pipeline with clock cycle time as 'Tp'. Let there be 'n' tasks to be completed in the pipelined processor. Now, the first instruction is going to take 'k' cycles to come out of the pipeline but the other 'n – 1' instructions will take only '1' cycle each, i.e, a total of 'n – 1' cycles. So, time taken to execute 'n' instructions in a pipelined processor.

$ET_{pipeline}$=k+n-1 cycles= (k+n-1)Tp

In the same case, for a non-pipelined processor, execution time of 'n' instructions will be:

$ET_{non\ pipeline}$= n*k*Tp

As the performance of a processor is inversely proportional to the execution time, we have,

S= $ET_{nonipeline}/ET_{pipeline}$= n*k*Tp/ (k+n-1)Tp

= (n*k)/(k+n-1)

49. Explain Arithmetic Pipeline with an example.

An arithmetic pipeline divides an arithmetic problem into various sub problems for execution in various pipeline segments. It is used for floating point operations, multiplication and various other computations.

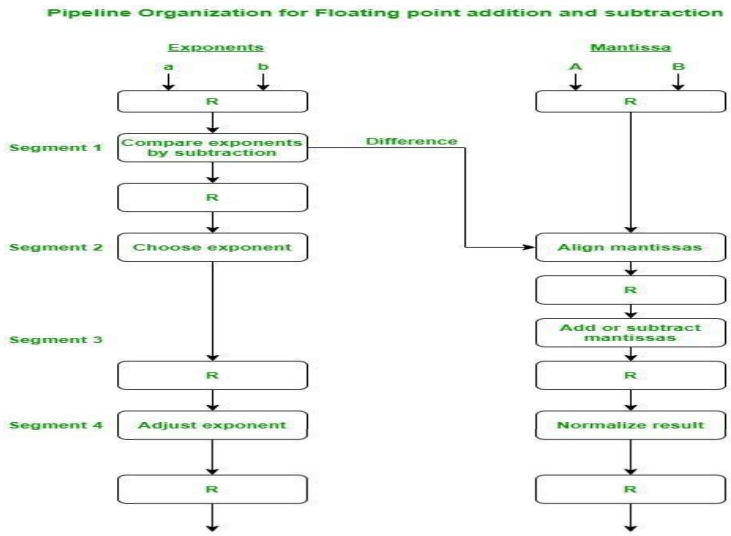
Example: Floating point addition using arithmetic pipeline

The following sub operations are performed:

- Compare the exponents.
- Align the mantissas.
- Add or subtract the mantissas.

For More Study Materials :https://www.keralanotes.com/

- Normalize the result



50. What is a pipeline hazards?

Pipeline hazards are situations that prevent the next instruction in the instruction stream from executing during its designated clock cycles. Any condition that causes a stall in the pipeline operations can be called a hazard.

51. Define structural, data, and control hazard.

Structural hazard

a. two different instructions use same h/w in same cycle – They arise from the resource conflict
b. HW cannot support all combination of instructions in overlapped fashion (single person to fold and put clothes away)

Data hazard

c. They arise when the instruction depends on the result of previous instruction.
d. must appear as if the instructions execute in correct order

Control hazard

e. one instruction affects which instruction is next
f. They arise when the pipelining of branches & other instructions that change the PC

52. List the techniques used for overcoming hazard.

Structural hazard: Stall, Use of additional hardware

Data Hazard: Stall, Data Forwarding, Code reordering

For More Study Materials :https://www.keralanotes.com/

Control Hazard: Stall, branch predict, delayed branch

53. Give examples of data hazard.

Read after Write (RAW): $Instr_J$ tries to read operand before $Instr_I$ writes it

Write after Read (WAR):$Instr_J$ tries to write operand *before* $Instr_I$ reads -Gets wrong operand

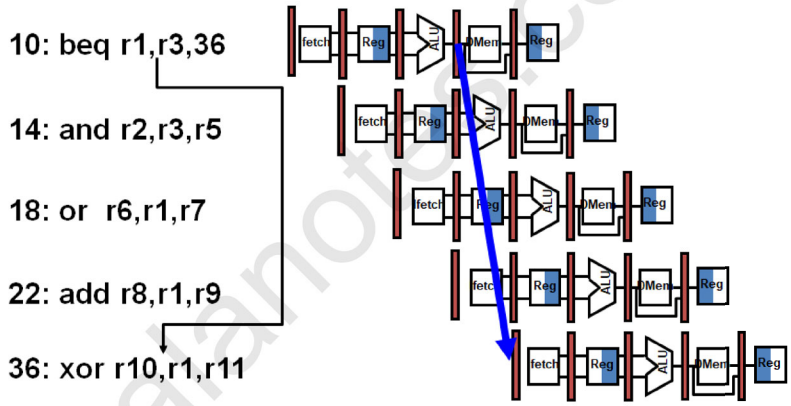
Write after Write (WAW): $Instr_J$ tries to write operand *before* $Instr_I$ writes it

- Leaves wrong result ($Instr_I$ not $Instr_J$)

- Read after Read (RAR).: It occurs when the instruction both read from the same register. No hazard occurs

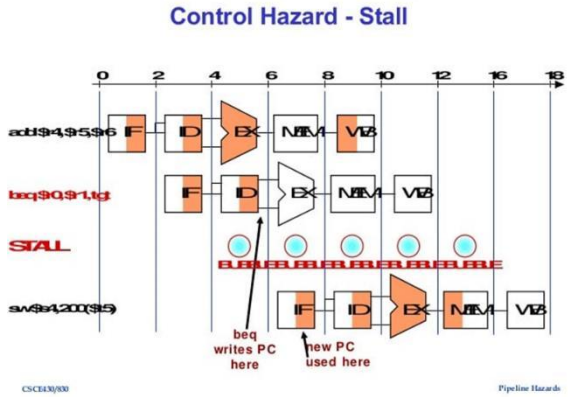
54. What is branch hazard? Describe the method for dealing with the branch hazard?

A control/branch hazard is when we need to find the destination of a branch, and can't fetch any new instructions until we know that destination.



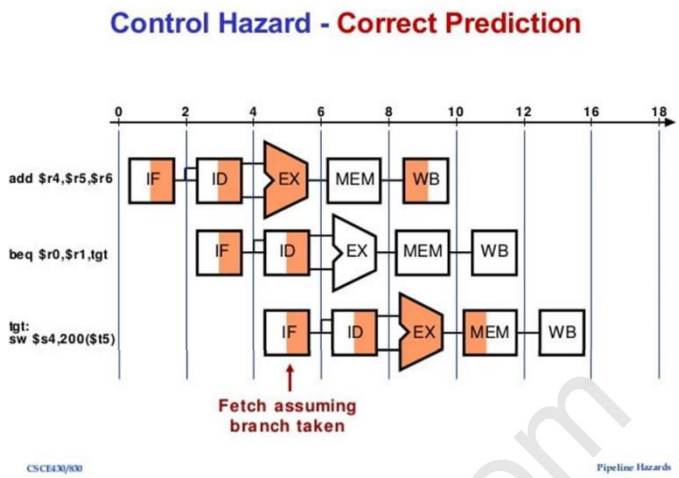
Solutions

- **Stall:** Stop loading instruction until result is available. Stalling for each branch is not practical



For More Study Materials :https://www.keralanotes.com/

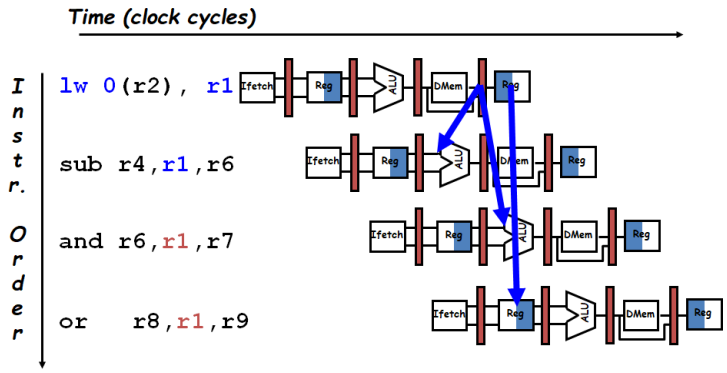
- **Predict:** Assume an outcome and continue fetching (undo if prediction is wrong)
  - Loses cycles only on mis prediction



- **Delayed Branch**: Specify in architecture that the instruction immediately following branch is always executed.

55. Explain the methods for dealing with data hazard.

Forwarding: It adds special circuitry to the pipeline. This method works because it takes less time for the required values to travel through a wire than it does for a pipeline segment to compute its result.



For More Study Materials :https://www.keralanotes.com/

Code reordeing: We need a special type of software to reorder code. We call this type of software a hardware-dependent compiler.

Stall insertion: it inserts one or more installs (no-op instructions) into the pipeline, which delays the execution of the current instruction until the required operand is written to the register file, but this method decreases pipeline efficiency and throughput.

56. Write the example of WAW hazard.

If instruction X tries to modify some data before it is written by instruction (X-1).