Module-3
CSEN 3104
Lecture 27

Dr. Debranjan Sarkar

Superscalar Architecture

Superscalar pipeline scheduling (Out-of-order issue)

- Lookahead window is used to reorder the instruction issues in order to shorten the total execution time
- By using the lookahead window, I5 can be decoded in advance because it is independent of all the other instructions
- I5 is fetched and decoded by the window, while I3 and I4 are decoded concurrently
- It is followed by issuing I6 and I1 at cycle 2, and I2 at cycle 3
- As the issue is out of order, the completion is also out of order
- Total execution time is reduced to 7 cycles with no idle stages
- Show the figures of the scheduling policy Out-of-order Issue

Superscalar pipeline scheduling

- In-order issue and completion is the simplest one to implement
- Unnecessary delays happen to maintain the program order
- So it is rarely used even in a conventional scalar processor
- In a multi-processor environment, this policy is attractive
- Allowing out-of-order completion can be found in both scalar and superscalar processors
- Some long-latency operations (e.g. Load and Floating point operations) can be hidden in out-of-order completion to achieve a better performance
- Output dependence and anti-dependence are the two relations that prevent out-of-order completion
- Out-of-order issue gives the processor more freedom to exploit parallelism, and thus pipeline efficiency is enhanced

Static Arithmetic Pipelines

- Arithmetic pipelines are mostly designed to perform fixed functions
- ALUs perform fixed-point and floating-point operations separately
- Fixed-point unit is also called the Integer Unit
- Floating-point unit may be built either as part of the CPU or on a separate coprocessor

Arithmetic Pipeline Stages

- Add, subtract, multiply, divide, squaring, square rooting, logarithm etc can be implemented with the basic add and shift operations
- A typical 3-stage floating-point adder includes
  - a 1st stage for exponent comparison and equalization (adder + shifting logic)
  - a 2nd stage for fraction addition (a high-speed carry look-ahead adder)
  - a 3rd stage for fraction normalization and exponent readjustment (shifter + adder)

Arithmetic Pipeline Stages

- Arithmetic or logical shifts can be implemented with shift registers
- High speed addition requires
  - Carry-propagation adder (CPA), or Show figure
  - Carry-save adder (CSA) Show figure
- In CPA, the carries propagate from the low end to the high end, using
  - Ripple carry propagation, or
  - Some carry look-ahead technique

Arithmetic Pipeline Stages

- In CSA, the carries are not allowed to propagate, but are saved in a carry vector
- Let X, Y, and Z are three n-bit input numbers, expressed as

$X = (x_{n-1}, x_{n-2}, \ldots, x_1, x_0)$ $Y = (y_{n-1}, y_{n-2}, \ldots, y_1, y_0)$ $Z = (z_{n-1}, z_{n-2}, \ldots, z_1, z_0)$

- The CSA performs bitwise operations simultaneously to produce two n-bit output numbers, denoted as

  $S^b = (0, S_{n-1}, S_{n-2}, \ldots, S_1, S_0)$ and

  $C = (C_n, C_{n-1}, \ldots, C_1, 0)$

- The input-output relationships (for i = 0, 1, 2, ..., n-1) are expressed as:

  $S_i = x_i$ XOR y_i XOR z_i

  $C_{i+1} = x_i y_i$ OR $y_i z_i$ OR $z_i x_i$

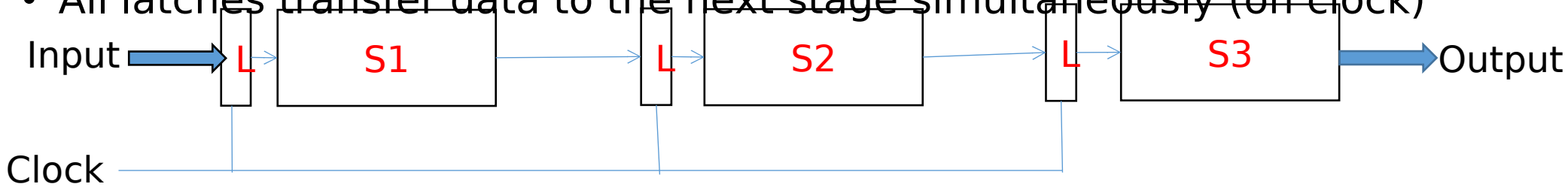
- $S = X + Y + Z = S^b + C$ (using CPA)

Multiply Pipeline Design

- Example of multiplication of two 8-bit integers A X B = P
- P is the 16-bit product
- Fixed point multiplication is the summation of 8 partial products (Show example)
- $P = A \times B = P_0 + P_1 + P_2 + P_3 + P_4 + P_5 + P_6 + P_7$
- P_j is (8+j) bits long with j trailing zeros
- The summation of 8 partial products is done with a Wallace tree of CSAs plus a CPA at the final stage (Show figure)
- S1: generates eight partial products
- S2: two levels of 4 CSAs taking eight numbers and producing four
- S3: two CSAs convert four numbers into two numbers
- S4: one CPA takes two numbers and result into one number

Principles of Pipelining (Recapitulation)

- Decomposes a sequential task into subtasks
- Each subtask is executed in a special dedicated stage
- These stages are connected with one another to form a pipe like structure.
- Instructions enter from one end and exit from another end
- Stages are pure combinational circuits for arithmetic or logic operations
- Result obtained from a stage is transferred to the next stage
- Final result is obtained after the instruction has passed through all the stages
- Stages are separated by high speed latches (or registers)
- All latches transfer data to the next stage simultaneously (on clock)



Multiply Pipeline Design

- For a maximum width of 16 bits, the CPA needs 4 gate levels of delay
- Each level of the CSA can be implemented with a 2 gate-level logic
- The delay of the first stage (S_1) also involves 2 gate levels
- So, all pipelines stages have an approximately equal amount of delay
- The matching of stage delays is important to determine the number of pipeline stages and the clock period
- If the delay of the CPA stage can be reduced to match

Thank you