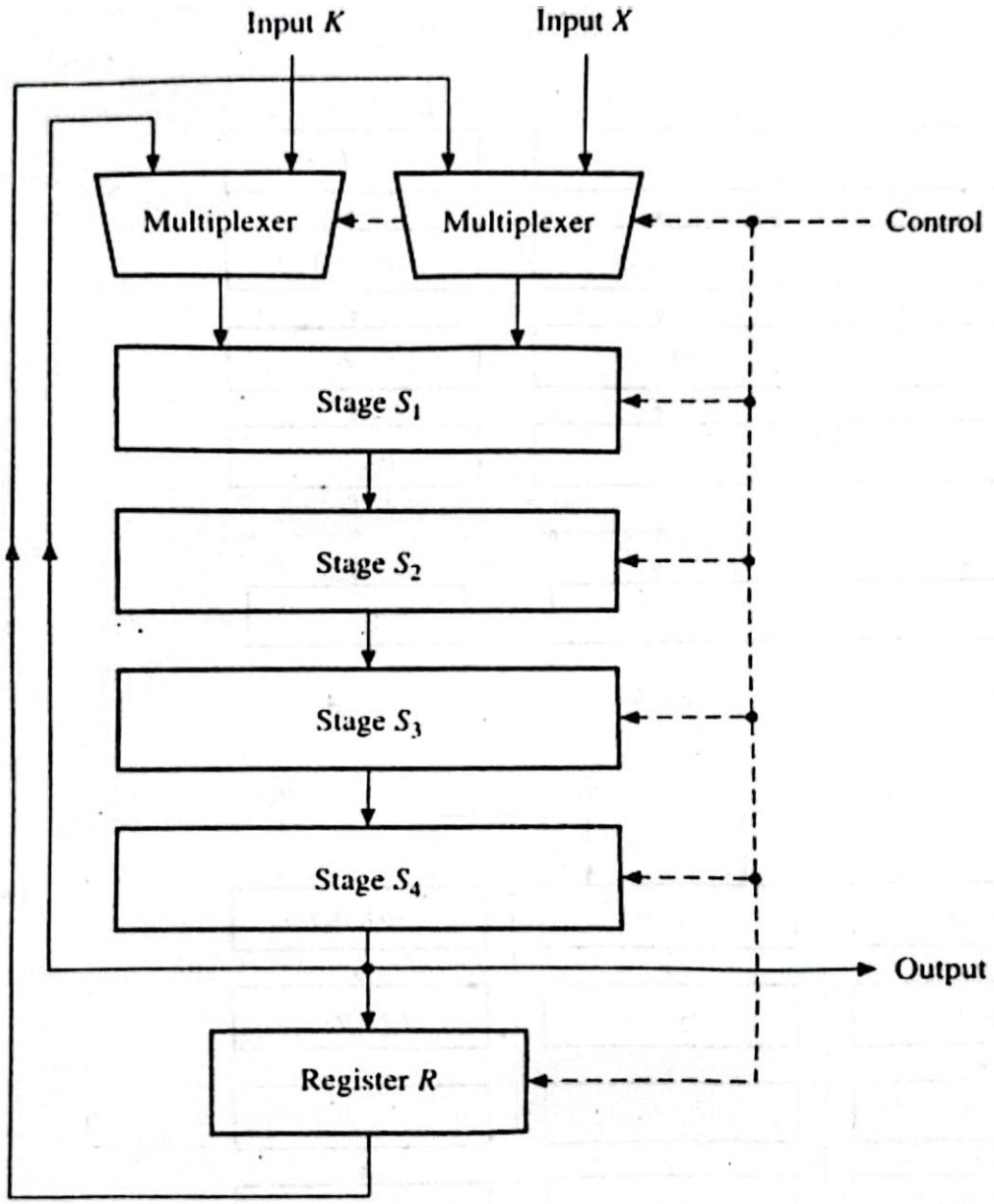
Lecture – 13

Feedback

- The usefulness of a pipeline processor can be enhanced by including feedback paths from the stage outputs to the primary inputs of the processor.

- It enables the results computed by certain stages to be used in subsequent calculations.

Summation by a Pipeline Processor

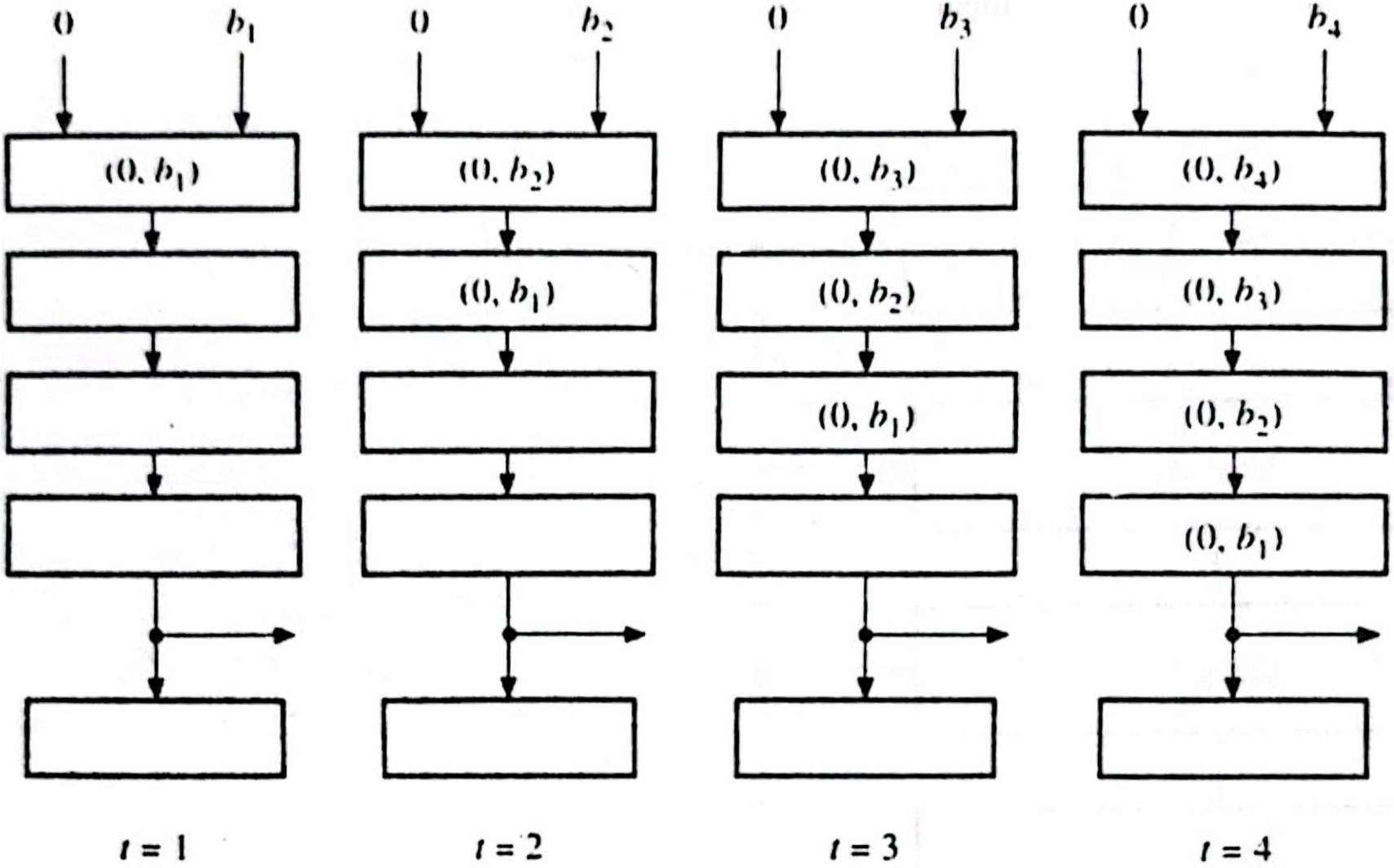


- Consider the problem of computing the sum of *N* floating-point numbers b_1, b_2, ……., b_N.

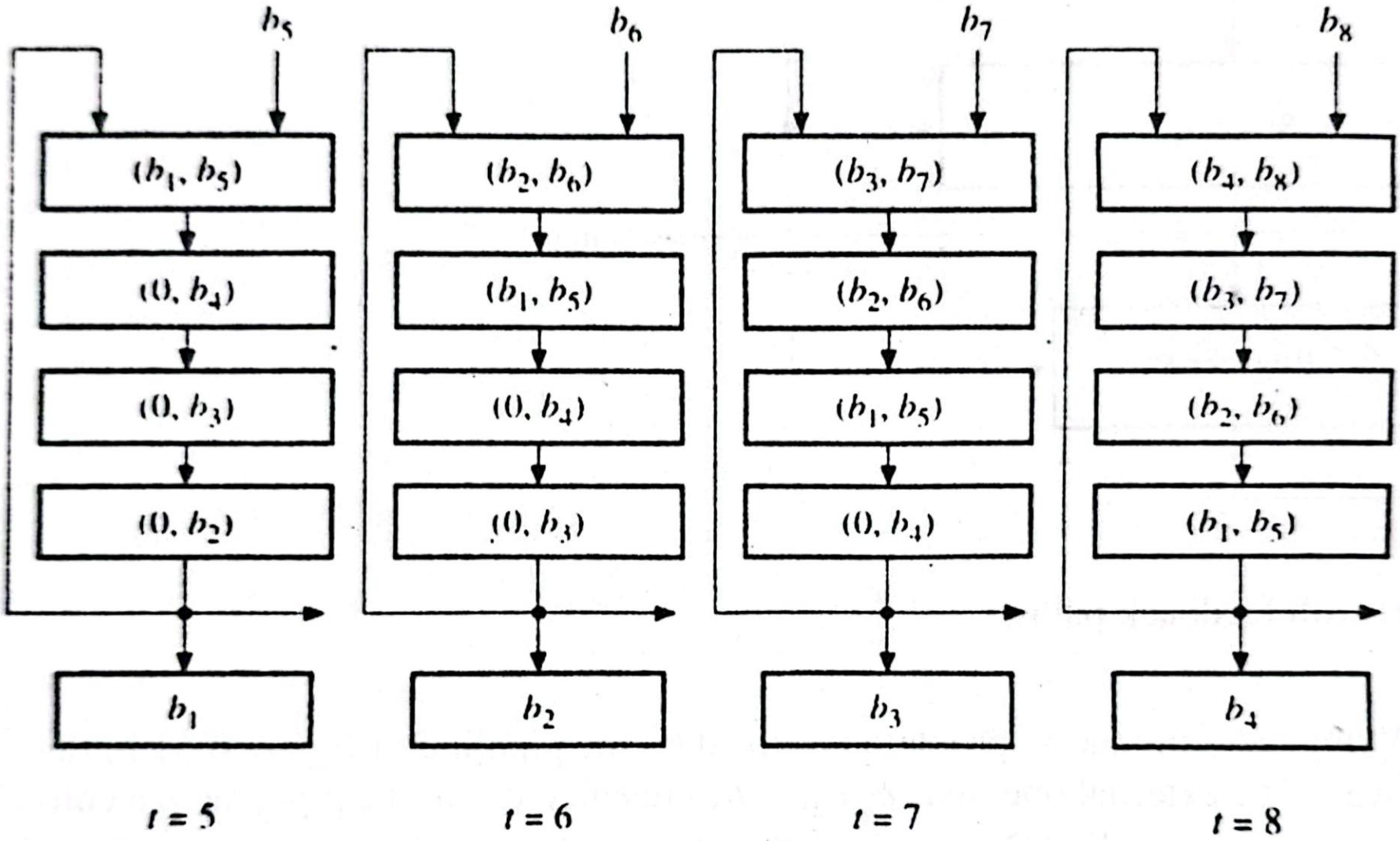
X = Obtained from a CPU register or a memory location.

K = all-0 words.

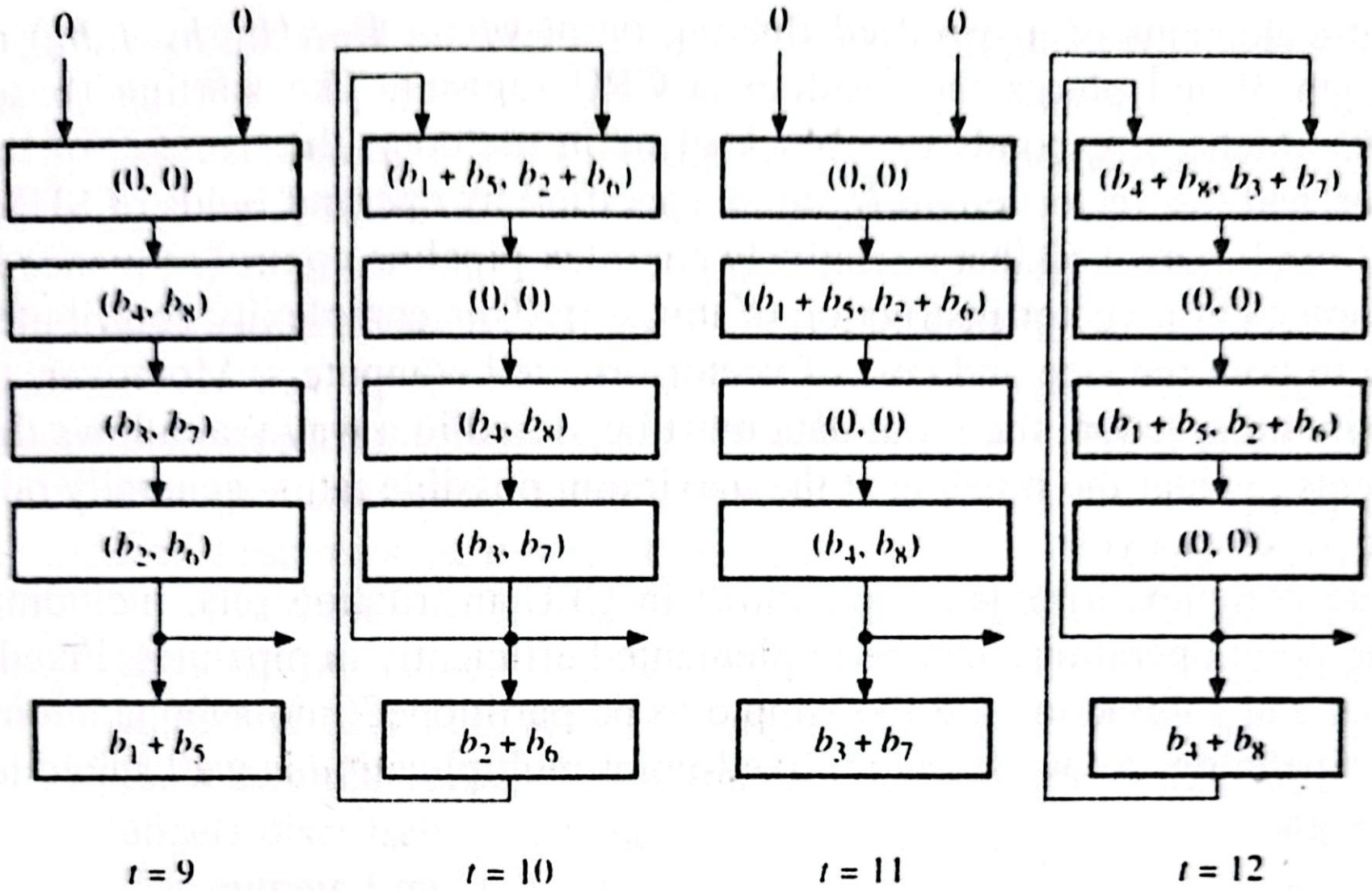
Summation by a Pipeline Processor



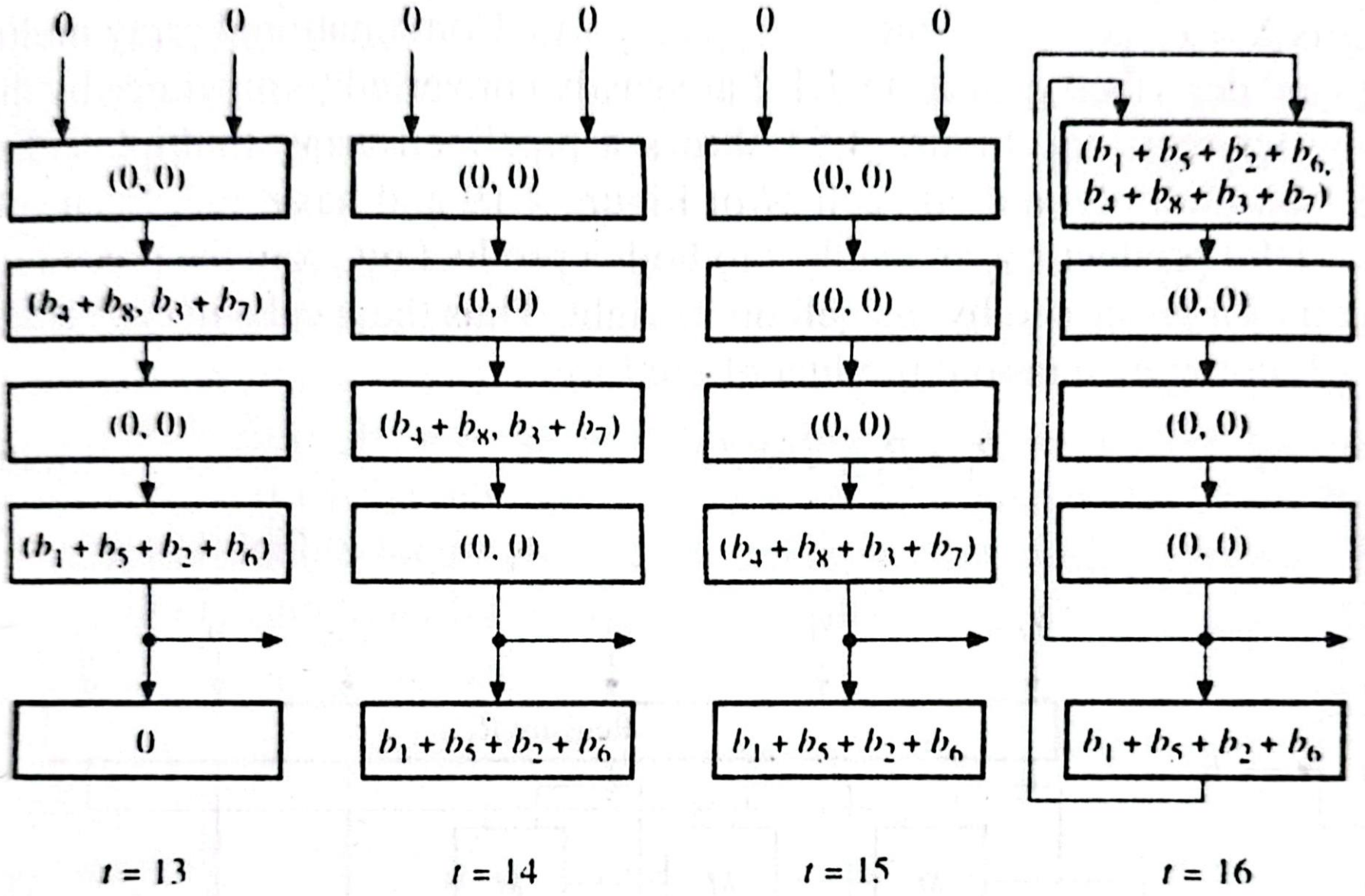
Summation by a Pipeline Processor



Summation by a Pipeline Processor



Summation by a Pipeline Processor

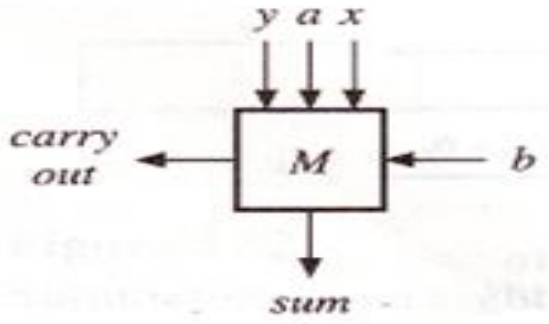


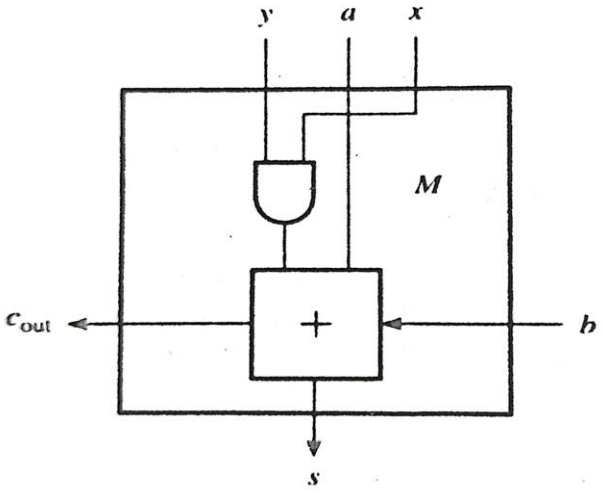
Summation by a Pipeline Processor

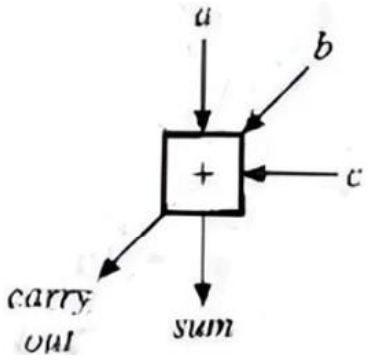
- The final result is produced four time periods later – at *t=20* in the case of *N=8*.
- The general case of *N (N>4)* operands, it takes *(N+12)T* time, where *T* is the pipeline's clock period.
- For nonpipelined adder it requires time *4NT* to compute the sum.
- So the speedup is *S(N) = 4N / (N+12).*
- For large N, S(4) ≈ 4.

Pipelined Multiplier

- A pipelined array multiplier employs the 1-bit multiply and add cell M.
- Each cell M computes product bit and add it to both a product bit preceding the stage and a carry bit generated by the cell on its right.



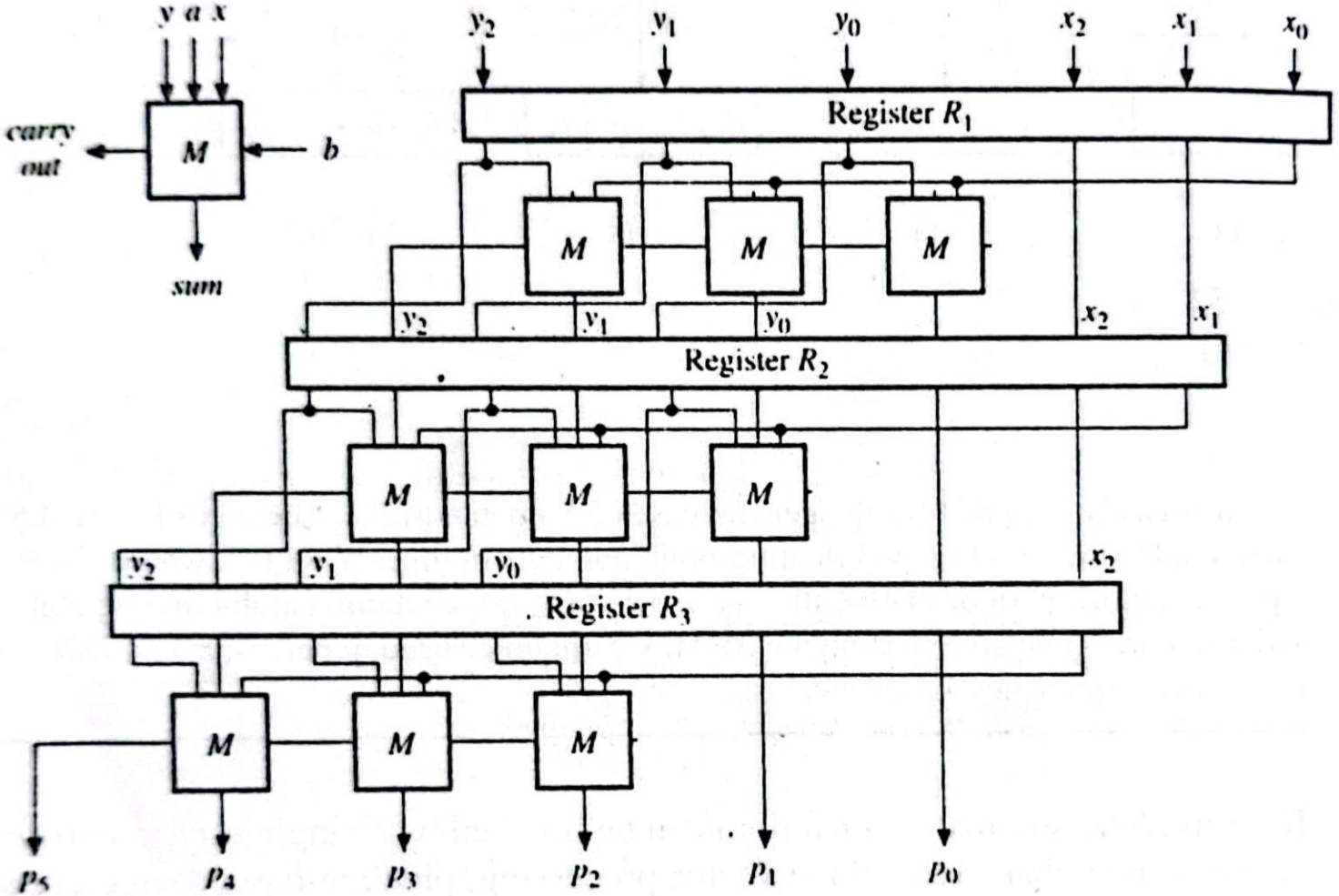




Pipelined Multiplier

- Multiply two n-bit fixed point numbers binary numbers

  $x = x_{n-1}, x_{n-2}, \ldots\ldots, x_1, x_0$ and $y = y_{n-1}, y_{n-2}, \ldots\ldots, y_1, y_0$



Here n =3.

$C_{out}S$ = a plus b plus xy

Pipelined Multiplier

- The circuit is designed for $n = 3$.
- The M cell computes $C_{out}S = a$ plus b plus xy
- The n cells in each stage S_i, $0 <= i <= n$ computes the partial product of the form $P_i = P_{i-1} + x_i 2^i Y$
- Buffer register R_i stores the partial product, the multiplicand Y and the unused multiplier bits.

Advantages of Pipelined Multiplier

- An n-stage multiplier pipeline can overlap the computation of n separate products.
- It generates a new result every clock cycle, after the delay of the pipeline.

Disadvantages of Pipelined Multiplier

- Slow speed of the carry propagation in each stage.
- The number of M cells needed in n^2.
- The capacity of all the buffer registers is approximately $3n^2$.
- It is costly in hardware.

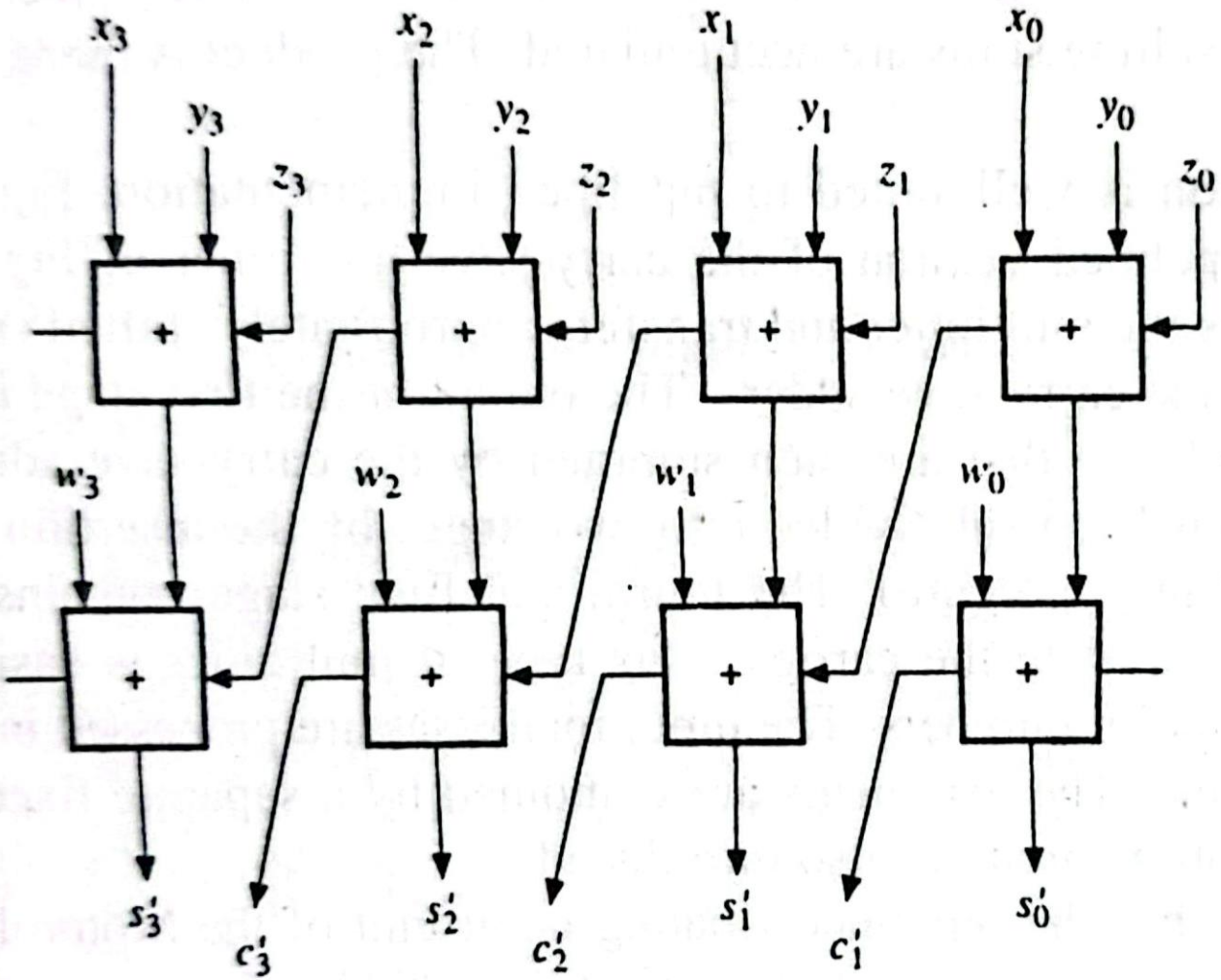
Carry Save Addition

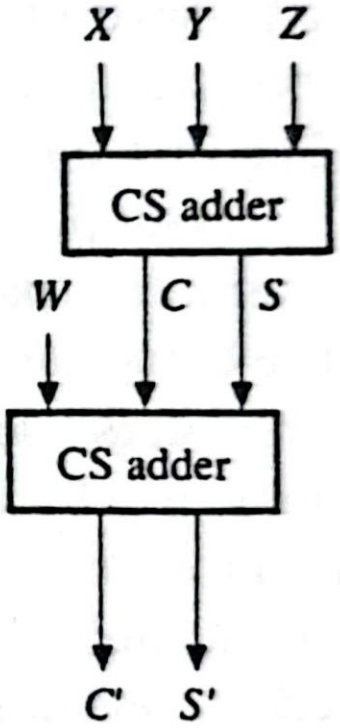
- An *n*-bit carry save adder consists of *n* disjoint full adders. It's input is three *n*-bit numbers to be added and the output consists of the *n*-sum bits forming a word *S* and the *n* carry bits forming a word *C*.

- Carry connections are shifted to the left correspond to normal carry propagation.

- To obtain the final result, *S* and *C* must be added by a conventional adder with carry propagation.

Carry Save Addition





Advantage of Carry Save Addition

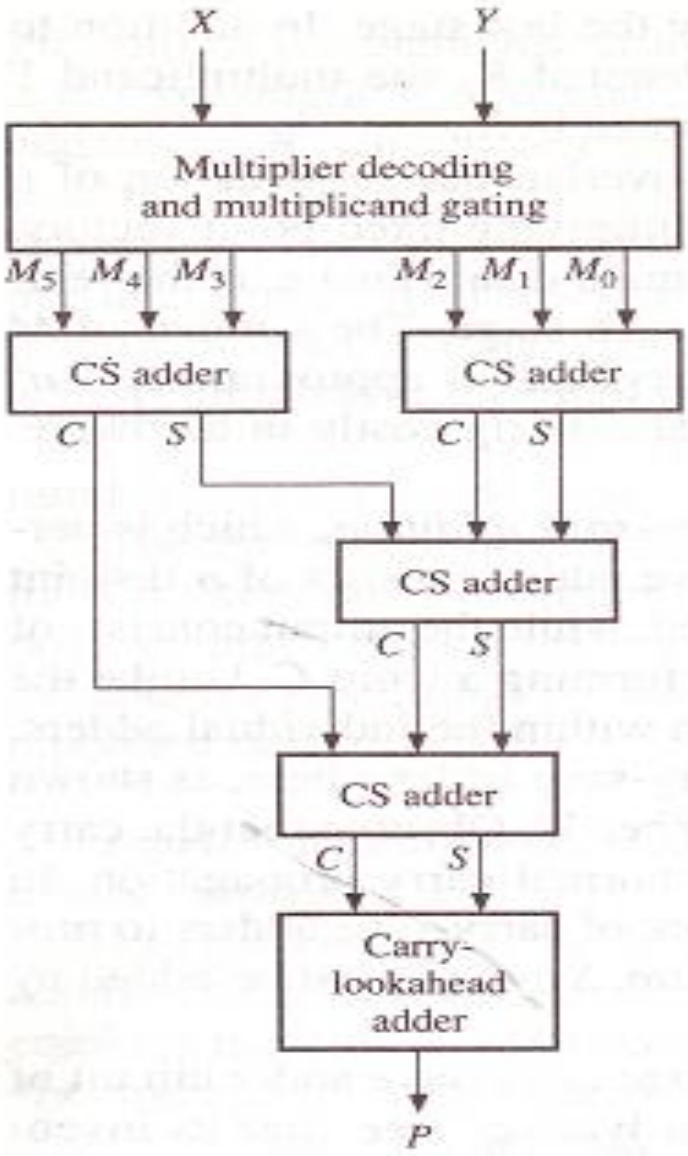
- All bits of *S* and *C* vectors are produced in a short, fixed amount of time.
- Carry propagation takes place only in the second row.
- Since all bits of *S* and *C* are available in parallel, a carry-lookahead adder can be used effectively to add the *S* and *C*.

Multiplication using Carry Save Adder

- Multiplication can be performed using a multi-stage carry-save adder circuit. This circuit is called ***Wallace Tree***.

- The inputs to the adder are *n*-terms of the form $M_i = x_i Y 2^k$ *where* M_i *represents multiplicand Y multiplied by the* i^{th} *multiplier bit weighted by power of 2.*
- This sum is computed by the carry-save adder tree, which produces a *2n*-bit sum and a *2n*-bit carry word.
- The final carry assimilation is performed by a carry-lookahead adder.

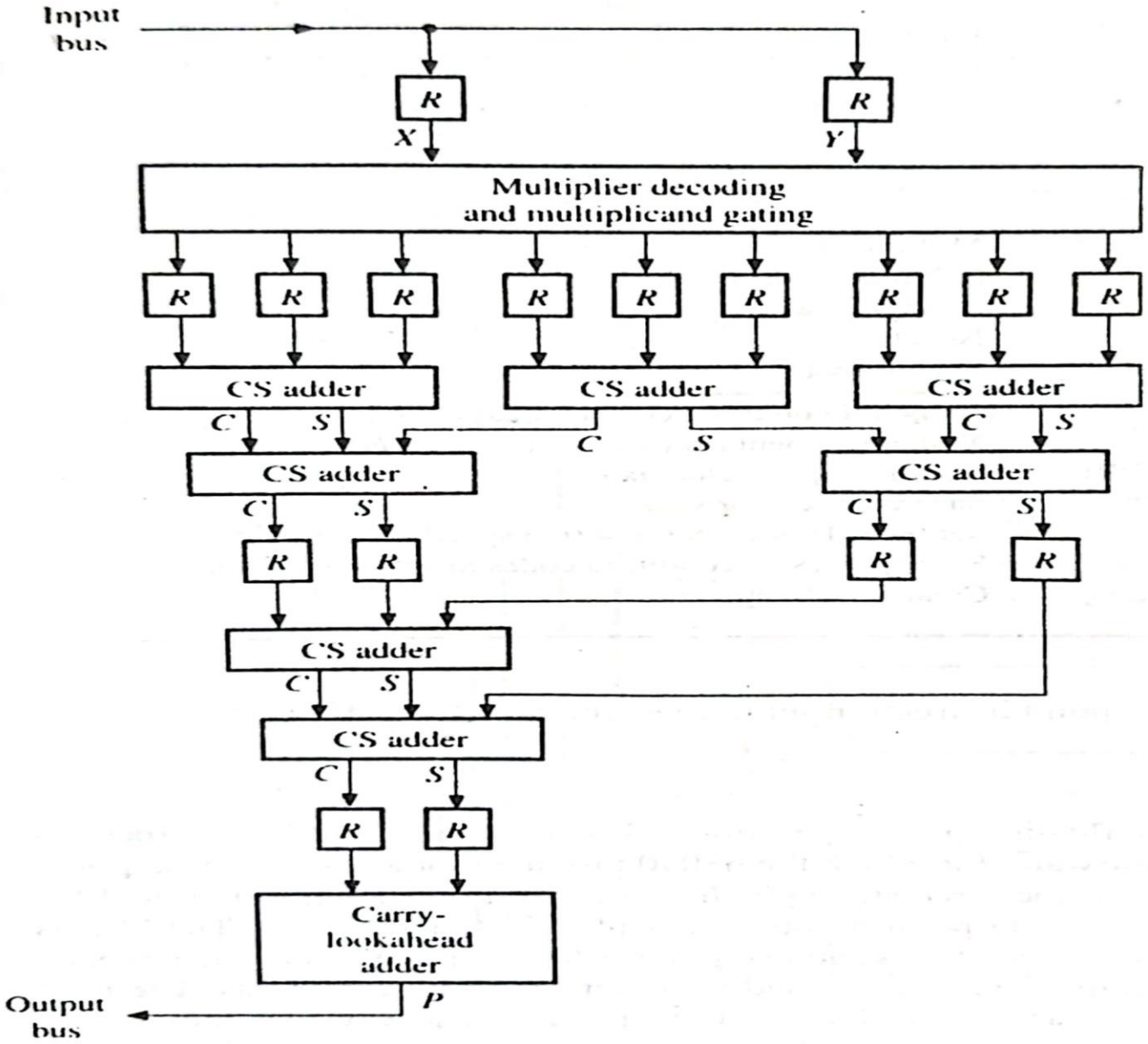
A Carry Save Multiplier



A Carry Save Multiplier

- It is practical for moderate values of *n*.
- For large *n*, the number of carry-save adders required can be excessive.
- Carry-save techniques can still be used, if the multiplier is partitioned into *k* *m*-bit segments. Only *m* terms M_i are generated and added via the carry-save adder circuits.
- The process is repeated *k* times, and the resulting sums are accumulated. The result is obtained after *k* iterations.

A Pipelined Carry Save Multiplier



Systolic Array

- It is formed by interconnecting a set of identical data-processing cells in uniform manner.
- Data words flow synchronously from cell to cell and each cell performs a small step in the overall operation of the array.
- It permits data to flow through the cells in several directions at once.
- One dimensional systolic array is therefore a kind of pipeline with identical stages.
- It can used to implement various complex arithmetic operations such as matrix multiplication

Two Dimensional Systolic Array Performing Matrix Multiplication

Let X be an $n \times n$ matrix of fixed-point or floating-point numbers defined by

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots x_{1,n} \\ x_{2,1} & x_{2,2} & \dots x_{2,n} \\ & & \dots \\ x_{n,1} & x_{n,2} & \dots x_{n,n} \end{bmatrix}$$

$$z_{i,j} = \sum_{k=1}^{n} x_{i,k} \times y_{k,j}$$

Two Dimensional Systolic Array Performing Matrix Multiplication

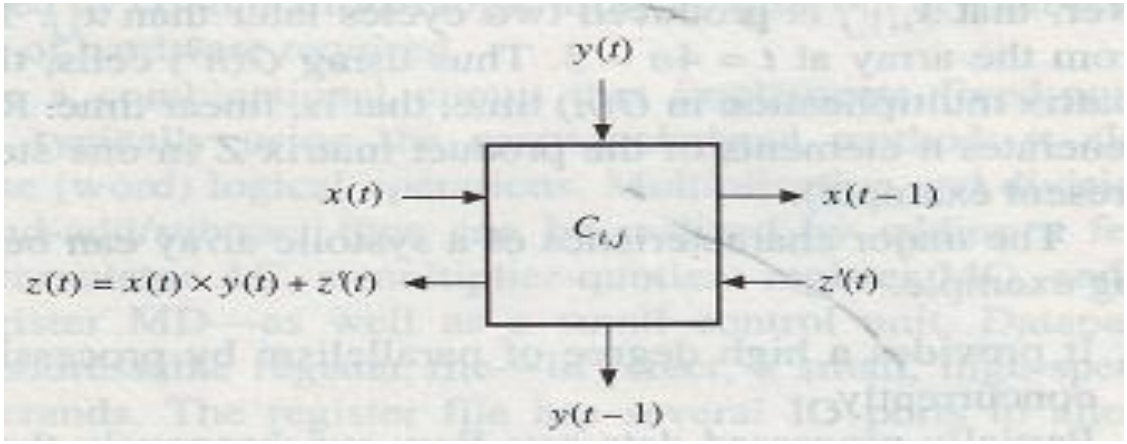
$$\begin{bmatrix} z_{11} & z_{12} & z_{13} \\ z_{21} & z_{22} & z_{23} \\ z_{31} & z_{32} & z_{33} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} * \begin{bmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \\ y_{31} & y_{32} & y_{33} \end{bmatrix}$$

$$z_{11} = x_{11}y_{11} + x_{12}y_{21} + x_{13}y_{31}$$

$$Z_{12} = x_{11}y_{12} + x_{12}y_{22} + x_{13}y_{32}$$

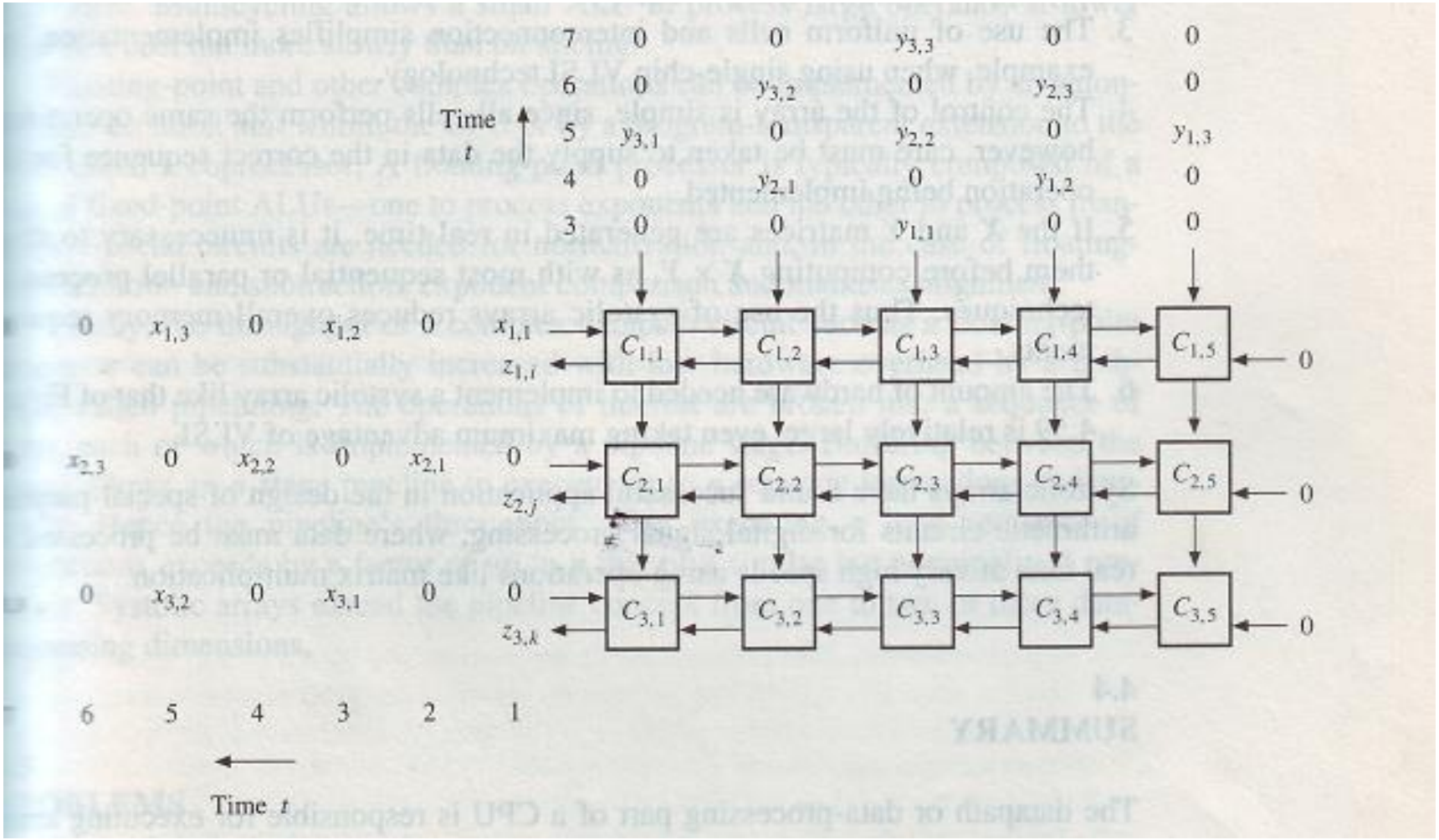
$$Z_{13} = x_{11}y_{13} + x_{12}y_{23} + x_{13}y_{33}$$

Two Dimensional Systolic Array Performing Matrix Multiplication



- Each cell $C_{i,j}$ of the matrix multiplier receives its *x* and *y*. It computes *z* and propagates its *x* and *y* input operands.

Two Dimensional Systolic Array Performing Matrix Multiplication



Two Dimensional Systolic Array Performing Matrix Multiplication

- The systolic matrix multiplier is constructed from *n(2n-1)* copies of $C_{i,j}$.
- The *x* and *y* operands are carefully ordered and separated by 0 so that the specific operand pairs $x_{i,k}$, $y_{k,j}$ meet at an appropriate cell of the array.
- The z's emerge from the left side of $C_{i,j}$.
- Each row of cells eventually issues the corresponding row of the matrix product *Z* from its left side.

Major Characteristics of A Systolic Array

- Provides a higher degree of parallelism.
- Partially processed data flow synchronously through the array in pipeline fashion but in several directions at once, with complete results eventually appearing at the array boundary.
- The use of uniform cells and interconnection simplifies the implementation, when using single chip.
- The control is simple. But care must be taken to supply the data in the correct sequence.
- X and Y need not to store before computing $X \times Y$.
- The amount of hardware needed to implement is relatively large.