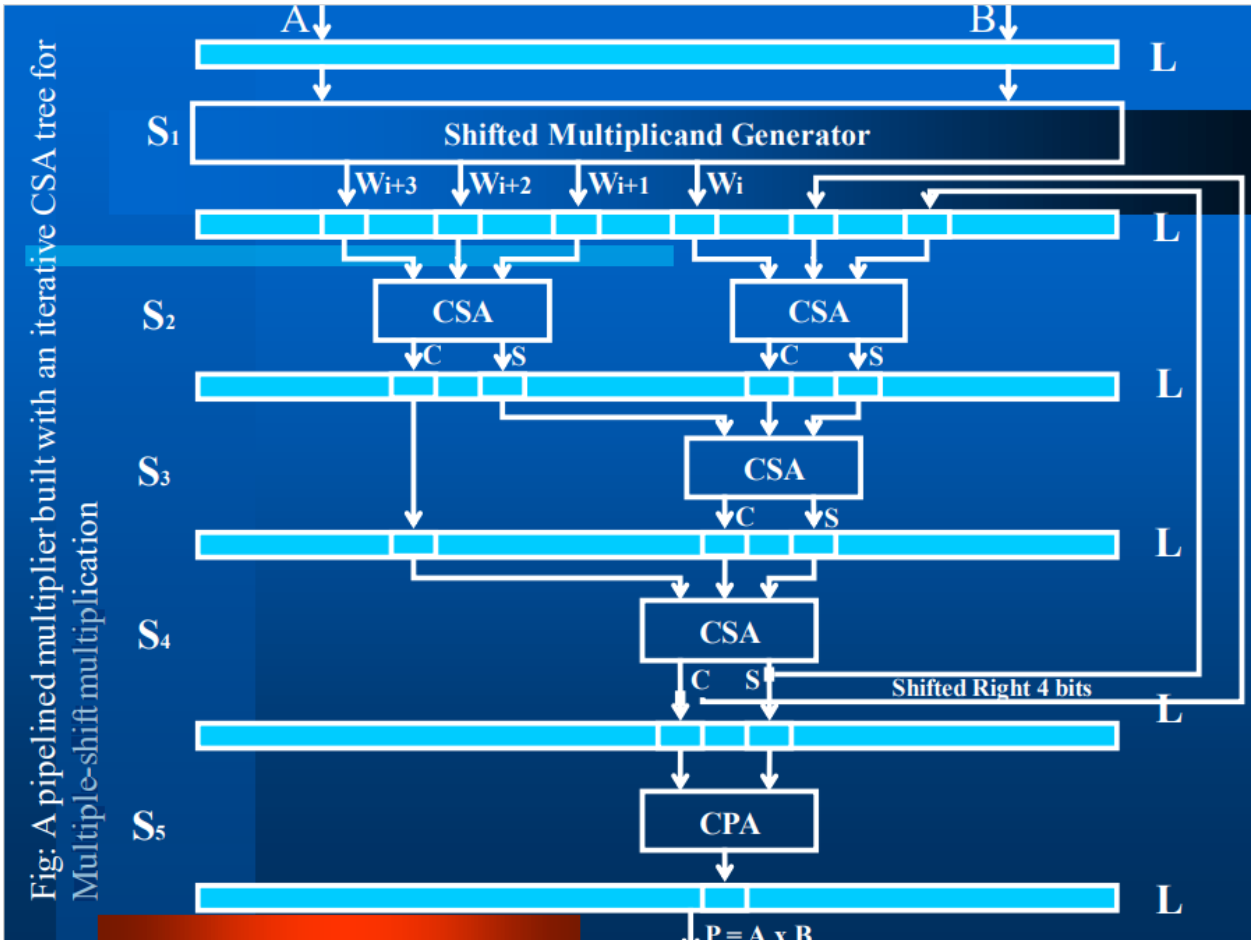
2. Draw and explain the structure of a pipelined multiplier built with an iterative CSA tree (6)



The pipeline multiplier with an iterative CSA tree for multiple shift multiplication is a type of multiplier that uses a combination of shifts and a carry-save adder (CSA) tree to multiply multiple signals in parallel. This type of multiplier is often used in high-performance applications, such as digital signal processing (DSP) and graphics processing units (GPUs).

The pipeline multiplier with an iterative CSA tree works by **first splitting the multiplicand into multiple segments**. Each segment is then shifted by a different amount and multiplied by the multiplier. The products of each multiplication are then **added together using a CSA tree**. The CSA tree is a type of adder that can add multiple numbers in parallel, which significantly speeds up the multiplication process.

The pipeline multiplier with an iterative CSA tree is divided into four stages:

1. **Shifted multiplicand generator**: This stage splits the multiplicand into multiple segments and shifts each segment by a different amount.
2. **Partial product generation**: This stage multiplies each segment of the multiplicand by the multiplier.
3. **CSA tree:** This stage adds together the products of each multiplication using a CSA tree.
4. **Carry propagation adder (CPA):** This stage converts the carry-save representation of the sum from the CSA tree to a regular binary representation.

The pipeline multiplier with an iterative CSA tree is a very efficient way to multiply multiple signals in parallel. It is often used in high-performance applications where speed is critical.

Here is a more detailed explanation of each stage of the pipeline multiplier with an iterative CSA tree:

Shifted multiplicand generator:

The shifted multiplicand **generator splits the multiplicand into multiple segments and shifts each segment by a different amount**. The number of segments and the amount of shift for each segment is determined by the width of the multiplier. For example, if the multiplier is 16 bits wide, then the multiplicand will be split into four segments, each of which will be shifted by 4 bits.

Partial product generation:

The **partial product generation stage multiplies each segment of the multiplicand by the multiplier**. This is done using a simple bit-by-bit multiplication. For each bit of the multiplicand, the corresponding bit of the multiplier is multiplied by the multiplicand bit. The product of each multiplication is then stored in a partial product register.

CSA tree:

The CSA tree is used **to add together the products of each multiplication**. The CSA tree is a type of adder that can add multiple numbers in parallel. The CSA tree works by first grouping the partial products into pairs. Each pair of partial products is then added together using a carry-save adder. The carry-save adder produces two outputs: a sum and a carry. The sum is stored in a register and the carry is passed to the next stage of the CSA tree.

The CSA tree is organized into multiple levels. The bottom level of the CSA tree contains the carry-save adders that add together the partial products. The **upper levels of the CSA tree add together the sums and carries from the lower levels**. The top level of the CSA tree produces the final sum of the multiplication.

Carry propagation adder (CPA):

The carry propagation adder is used to convert the carry-save representation of the sum from the CSA tree to a regular binary representation. The CPA **works by propagating the carry bits from one stage to the next until all of the carry bits have been propagated.** The final output of the CPA is the regular binary representation of the sum.

The pipeline multiplier with an iterative CSA tree is a very efficient way to multiply multiple signals in parallel. It is often used in high-performance applications where speed is critical.