DIGITAL SYSTEM DESIGN

Building Block Circuits

- Rather than building systems at the gate level, often digital systems are constructed from higher level, but still basic, building block circuits.
- Multiplexers, decoders, flip-flops, registers, and counters are examples of building blocks, which are subcircuits from which complex circuits can be constructed.
- For many larger systems, the circuitry required can often be divided into two Sub-systems: the **datapath circuit; and the control circuit.**
- The *datapath* circuit is used to store and manipulate data and to transfer data from one part of the system to another.
- Datapath circuits can comprise of building blocks such as registers, shift registers, counters, multiplexers, decoders, etc.

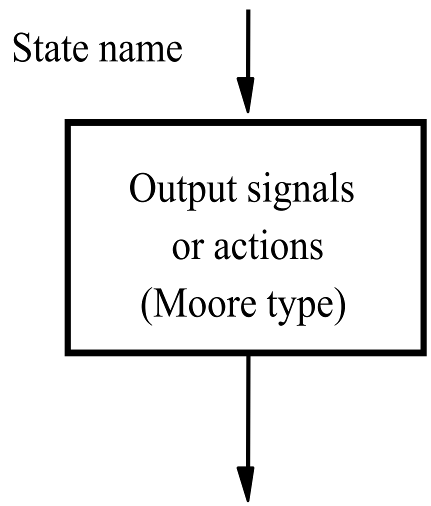
Building Block Circuits

- The *control* circuit, usually an FSM, controls the operation of the datapath circuit.
- In many applications, it is useful to be able to prevent the data stored in a flip-flop from changing when an active clock edge occurs.
- A simple example of the division of the data path and the control path can be illustrated using a flip-flop with an enable input.
- The data path consists of the flip-flop and its input, and the control path consists of the enable input.
- The two paths exist independently of each other with the enable (control path) controlling the flow of the data into the flip-flop.
- It is also useful to be able to inhibit the shifting operation in a shift register by using an enable input.

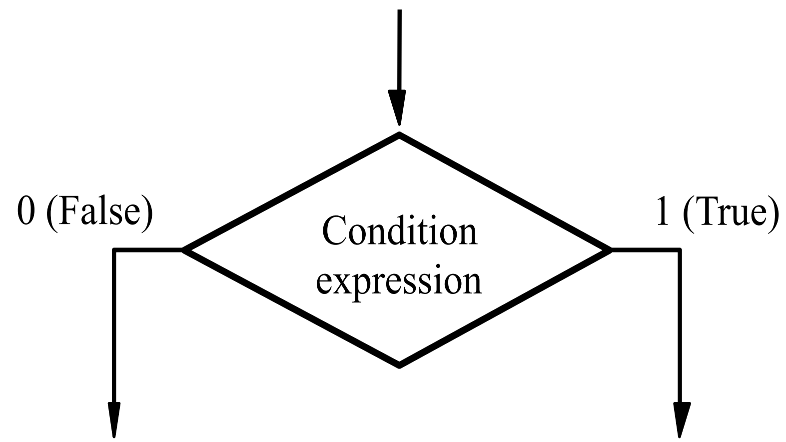
Algorithmic State Machine (ASM) charts

- State diagrams are not convenient to describe the behavior of large state machines
- ASM charts are used to describe large machines
  - It is a type of flow chart
  - Represents state transitions
  - Represent generated outputs for an ASM
- ASM charts have three types of elements
  - State box
  - Decision box
  - Conditional output box

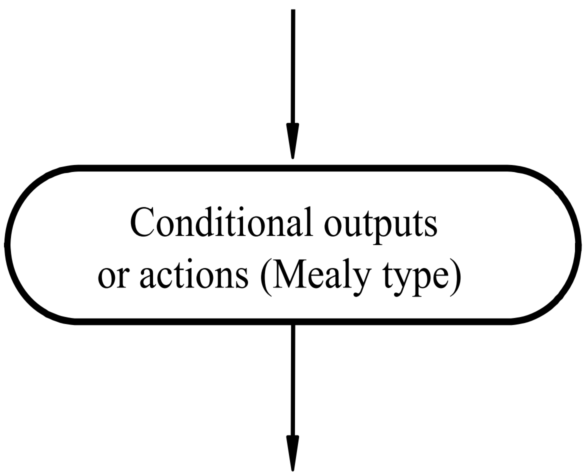
Elements used in ASM charts



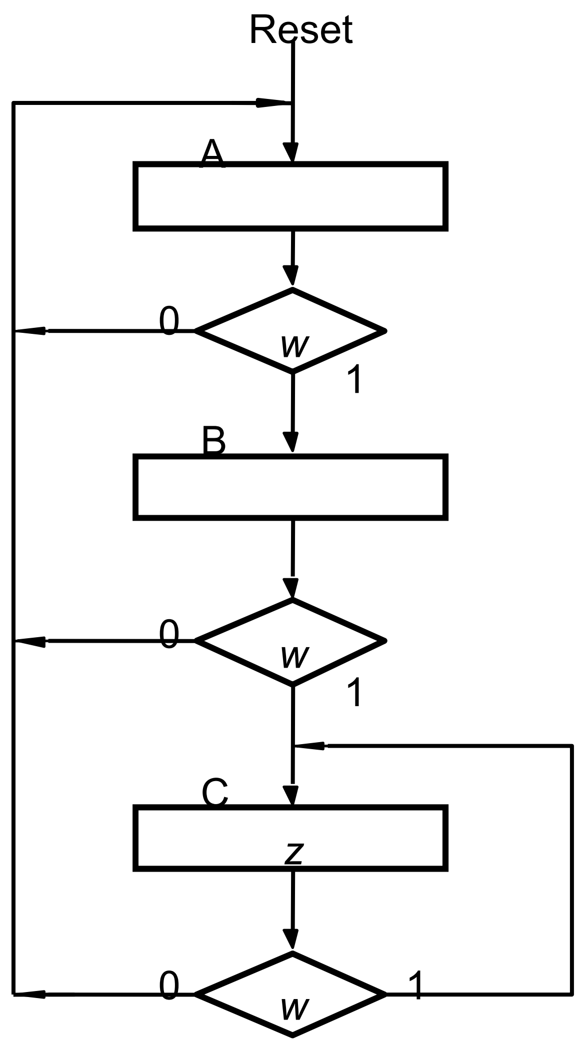
(a) State box



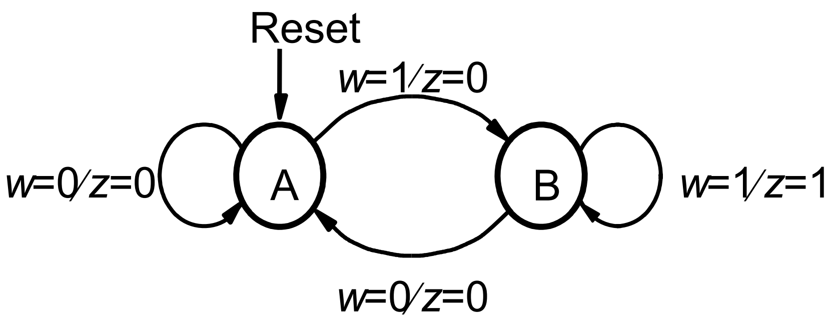
(b) Decision box

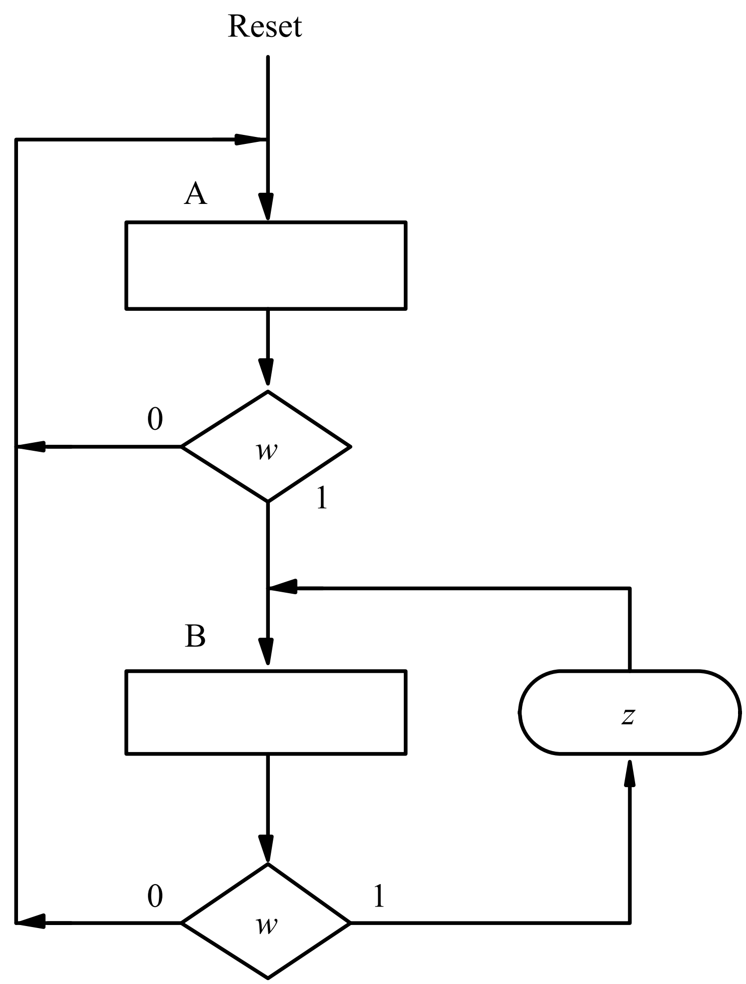


ASM chart for a simple FSM



State Diagram and its corresponding ASM chart





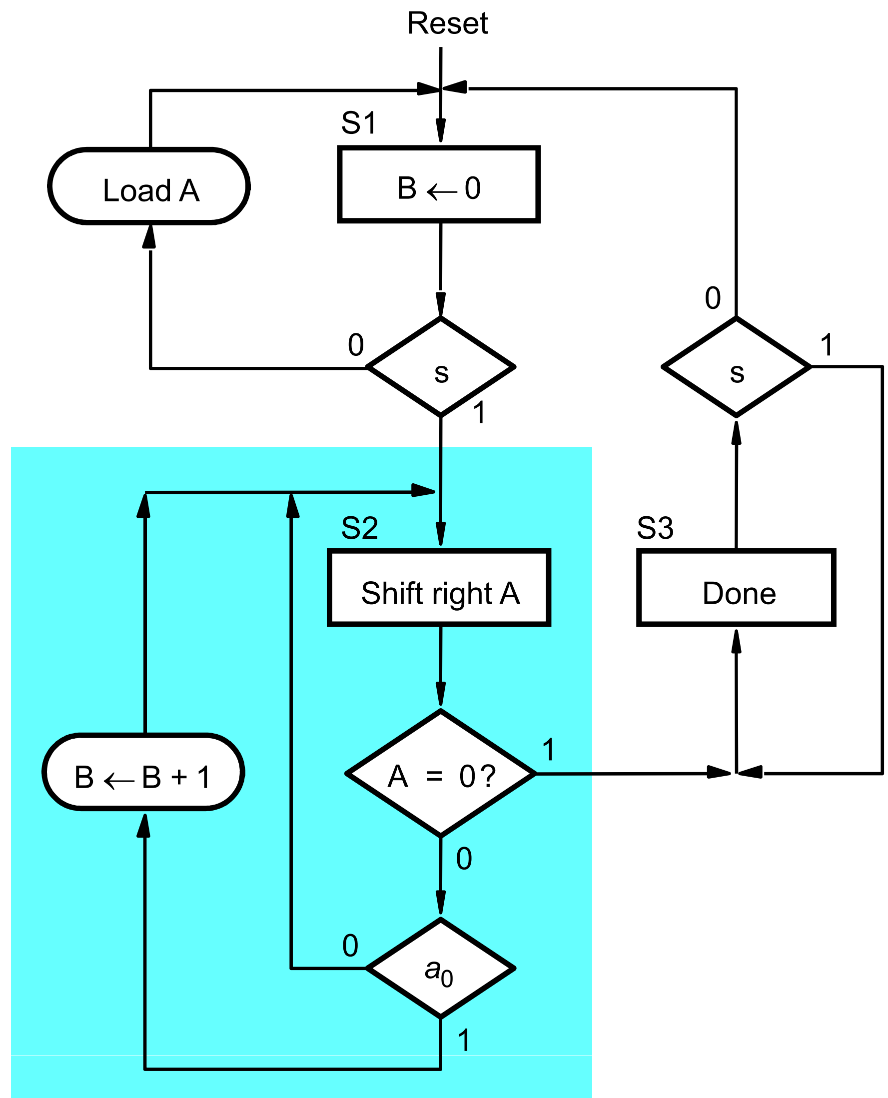
Design Example: A Bit-Counting Circuit

- Using the concepts of the ASM and the separate data and control circuits we can implement fairly complex systems.
- Suppose we wish to count the number of bits in a register that have the value 1.
- Assume that the value A is stored in a register that can shift its contents in the left to-right direction.
- Pseudo-code for the bit counter.

```
B=0;
while A ≠ 0  do
        if a0 = 1   then
                B=B+1;
        End if;
        Right-shift A ;
End while;
```

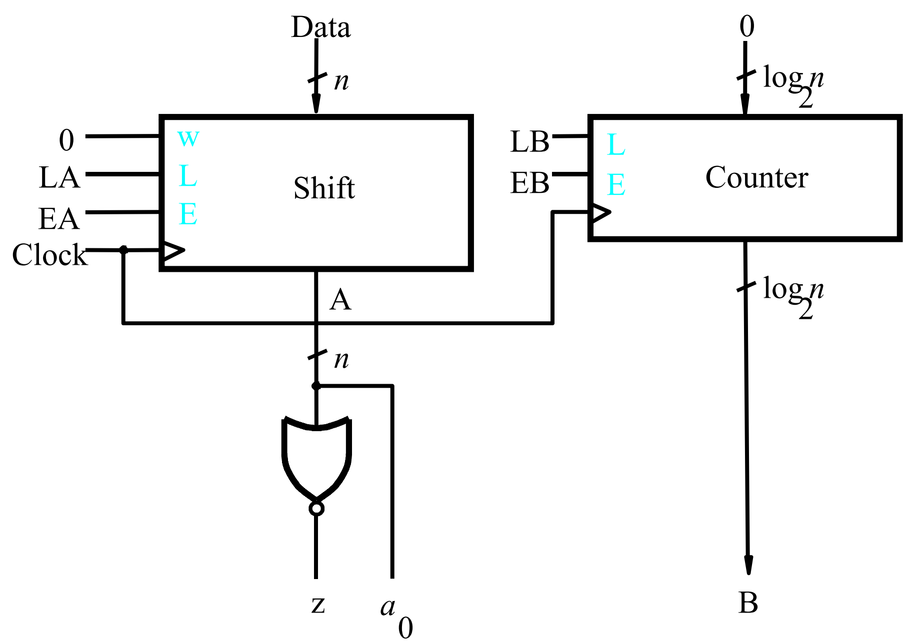
ASM chart for the pseudo-code.

- s: input signal that indicates if A has been loaded
- We can assume that the same clock signal controls the changes in the state of the machine and changes in A and B. Therefore in state S2, the decision box which tests whether A=0, occurs simultaneously with the box that checks the value of a0.
- If A=0, then the FSM will change to state S3 on the next clock edge (this also shifts A, which has no effect because A is already 0).
- On the other hand, if A=0, then the FSM does not change to S3 but remains in S2. At the same time A is shifted, and B is incremented if a0 has the value 1.

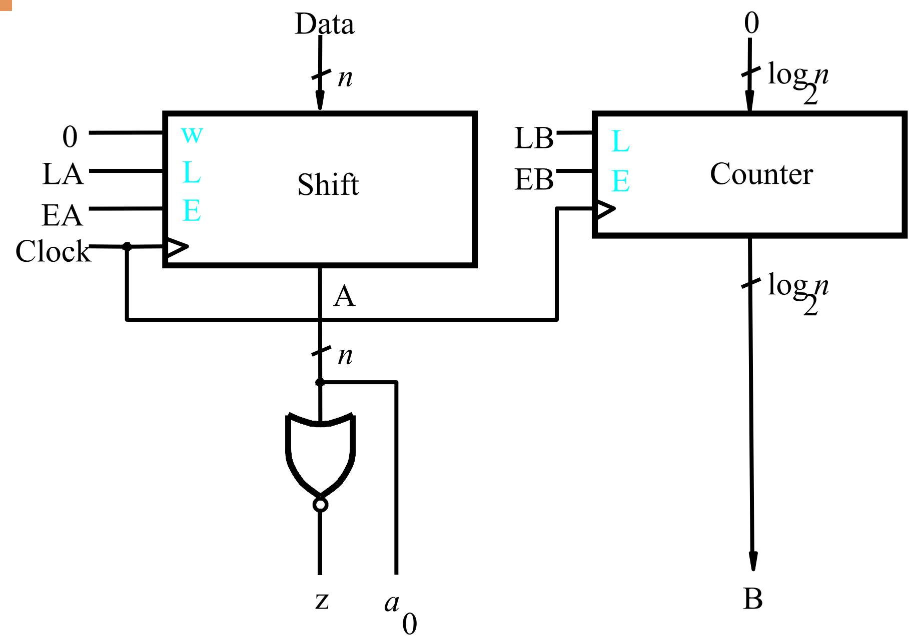


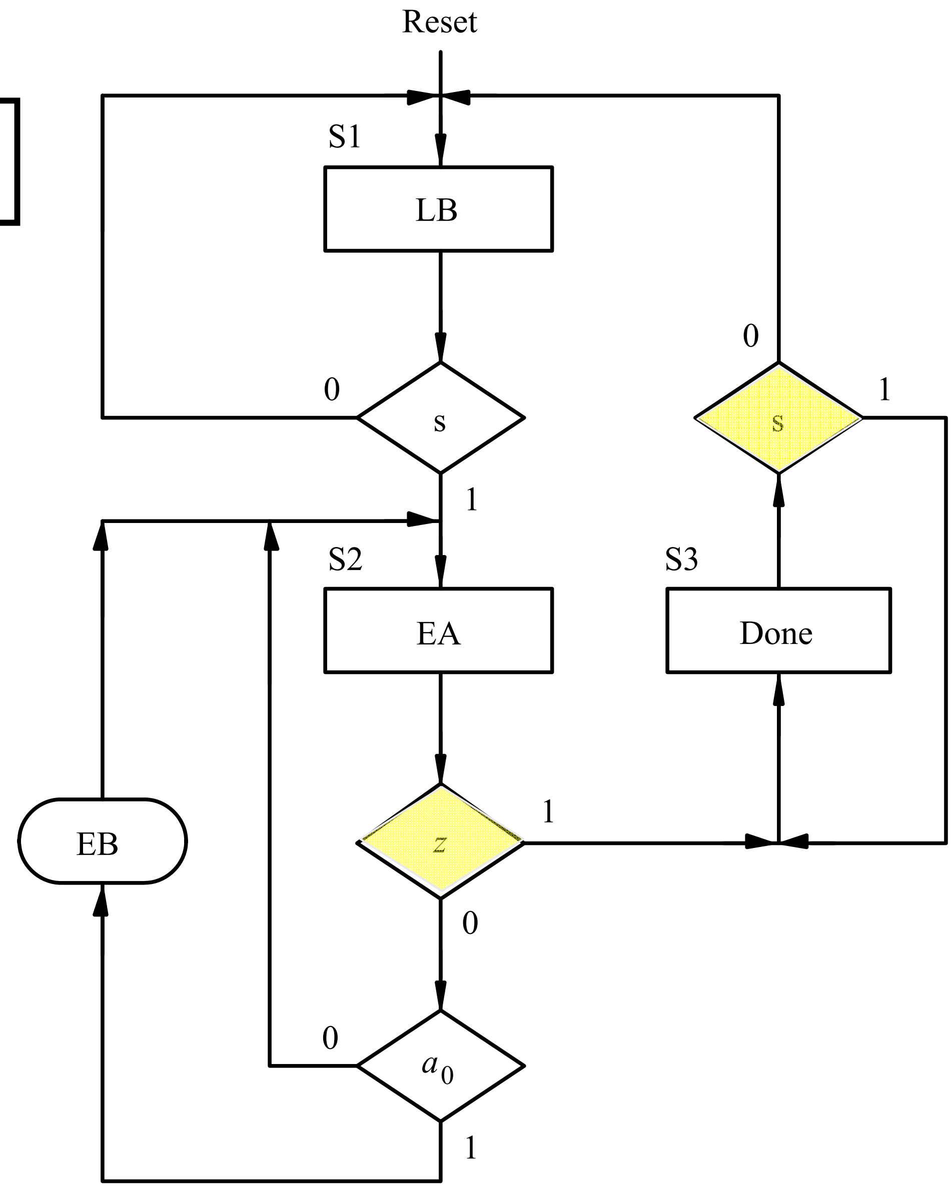
A Bit-Counting Circuit (data-path)

- For the data-path circuit a shift register which shifts left to-
- right is required to implement A.
- It must have the parallel load capability and an enable input since shifting should occur only in state S2.
- In addition, a counter is needed for B, and it needs a parallel-load capability to initialize the count to 0 in state S1.



ASM chart for the bit counter control circuit

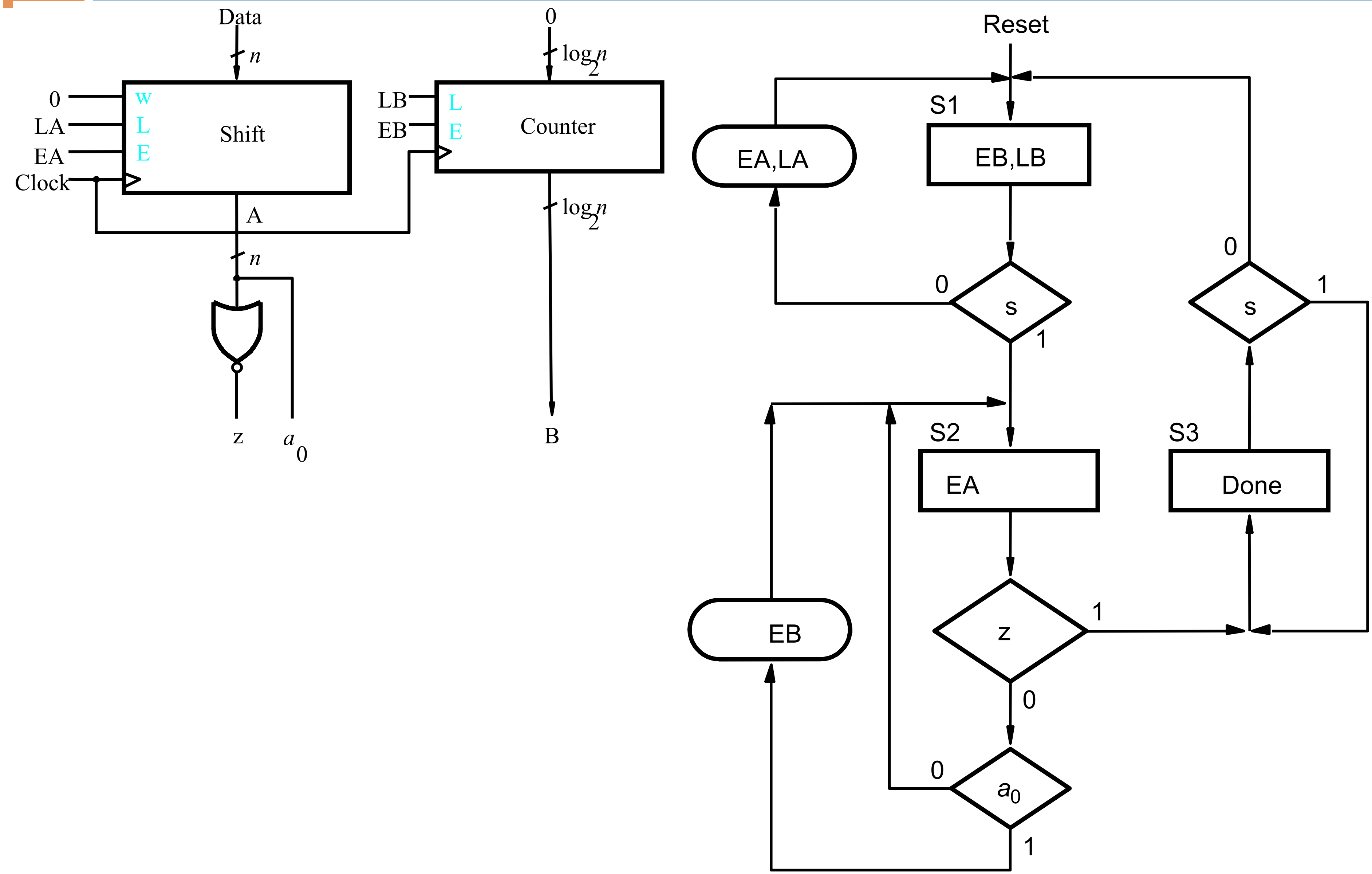




s: A is ready

z: =1 when A = 0 (filled it)

ASM chart for the bit counter control circuit



Shift-And-Add Multiplier

Decimal

$$
\begin{array}{r}
13 \\
\times\ 11 \\
\hline
13 \\
13 \\
\hline
143
\end{array}
$$

Binary

$$
\begin{array}{r l}
1\ 1\ 0\ 1 & A\text{: Multiplicand} \\
\times\ 1\ 0\ 1\ 1 & B\text{: Multiplier} \\
\hline
1\ 1\ 0\ 1 & \\
1\ 1\ 0\ 1\ & \\
0\ 0\ 0\ 0\ & \\
1\ 1\ 0\ 1\ & \\
\hline
1\ 0\ 0\ 0\ \ 1\ 1\ 1\ 1 & P = \text{Product}
\end{array}
$$

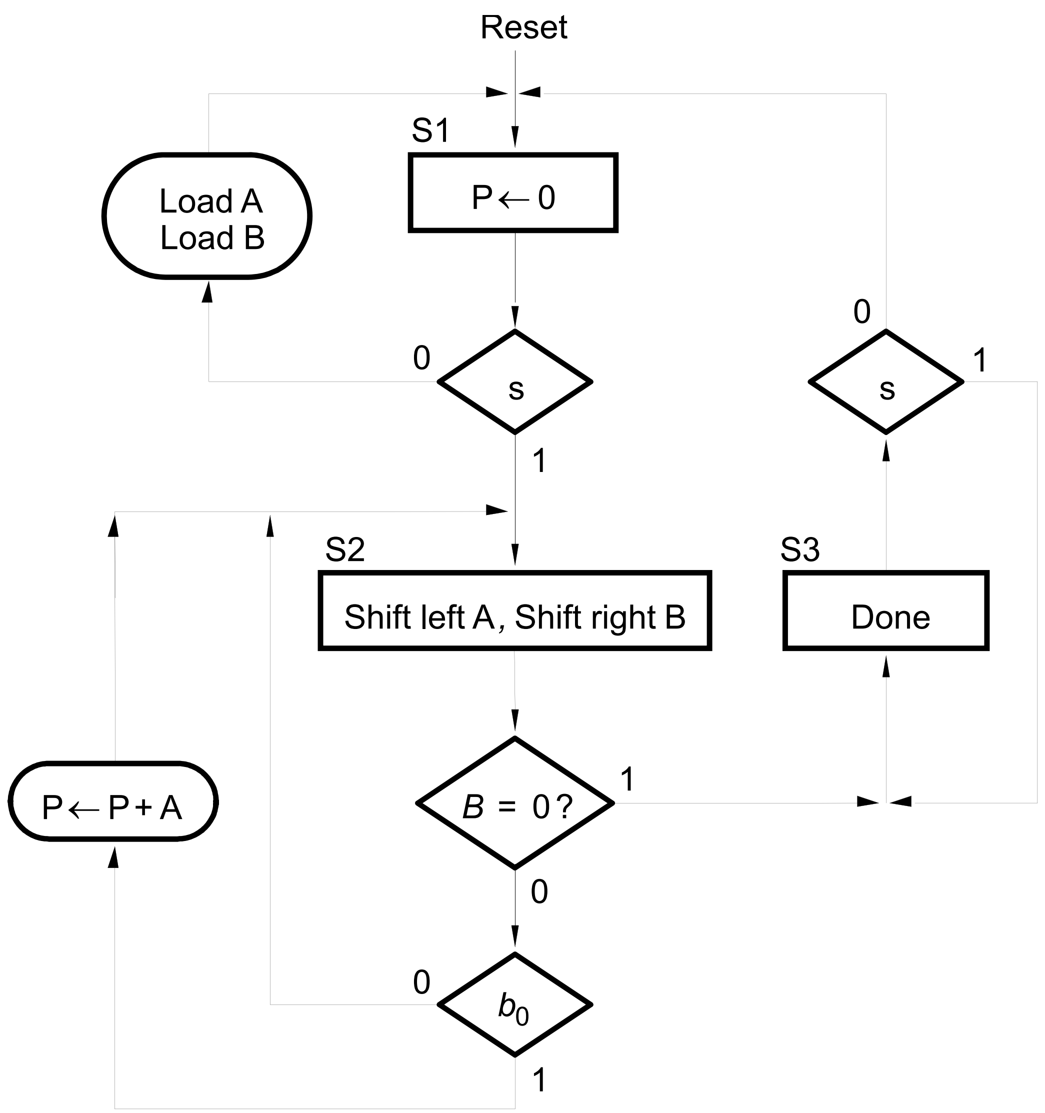
Manual method

An algorithm for multiplication.

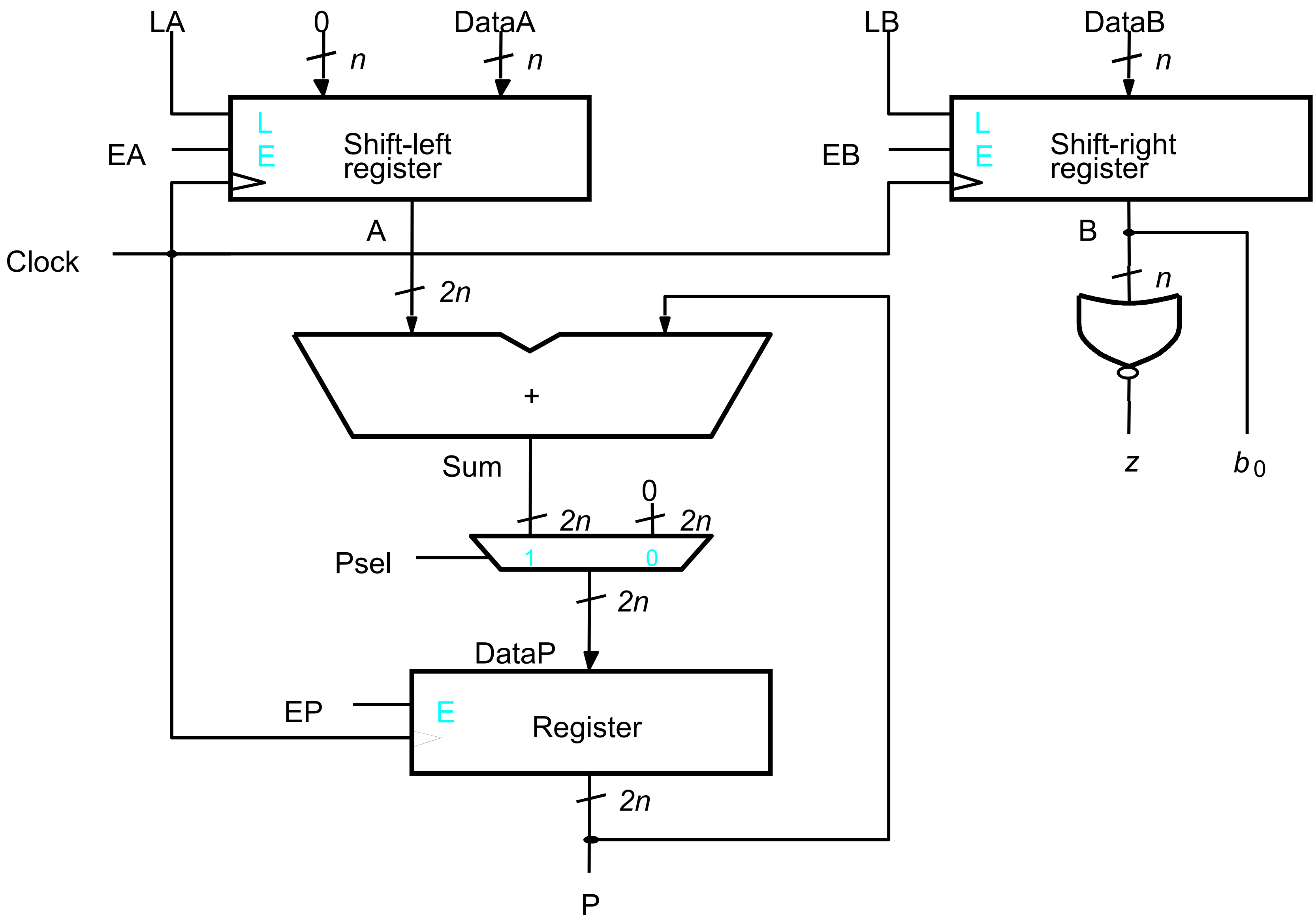
$P = 0$;
for $i = 0$ to $n - 1$ do
  if $b_i = 1$ then
    $P = P + A$;
  end if;
  Left-shift A ;
end for;

ASM chart for the multiplier

$P = 0$;
for $i = 0$ to $n - 1$ do
    if $b_i = 1$ then
        $P = P + A$;
    end if;
    Left-shift A ;
end for;



Datapath circuit for the multiplier.



ASM chart for the multiplier control circuit.