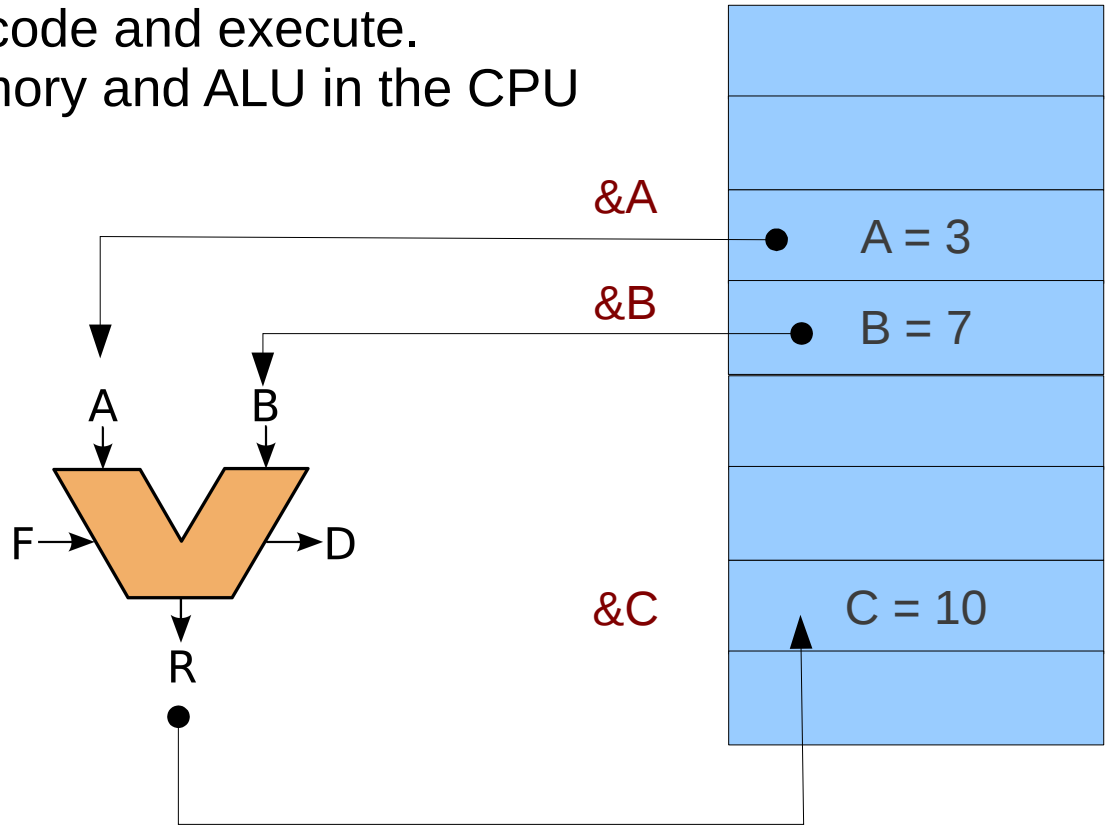
#1 Complete the following C code and execute.
Explain what happens in Memory and ALU in the CPU

```
int A, B, C;
int * p;

A = 3;
B = 7;
C = A + B;
A = 2;
B = 5;
C = A + B;
A = 3;
B = 3;
C = A + B;
```




```
printf("address of A = %x, data of A = %d \n", &A, A);
printf("address of B = %x, data of B = %d \n", &B, B);
printf("address of C = %x, data of C = %d \n", &C, C);

p = &A; printf("address of A = %x, data of A = %d \n", p, *p);
p = &B; printf("address of A = %x, data of A = %d \n", p, *p);
p = &B; printf("address of A = %x, data of A = %d \n", p, *p);
```

#2 Design the Hardware which will do the same thing in #1.

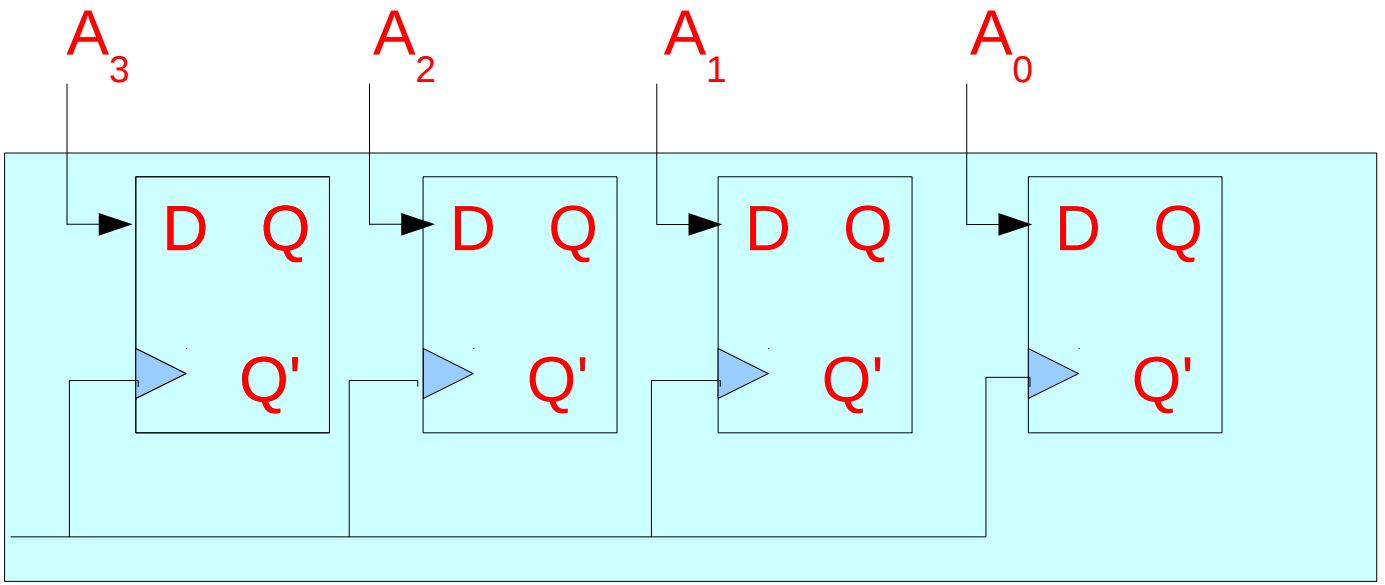
$A \rightarrow A_3A_2A_1A_0$

$B \rightarrow B_3B_2B_1B_0$

$C \rightarrow C_3C_2C_1C_0$

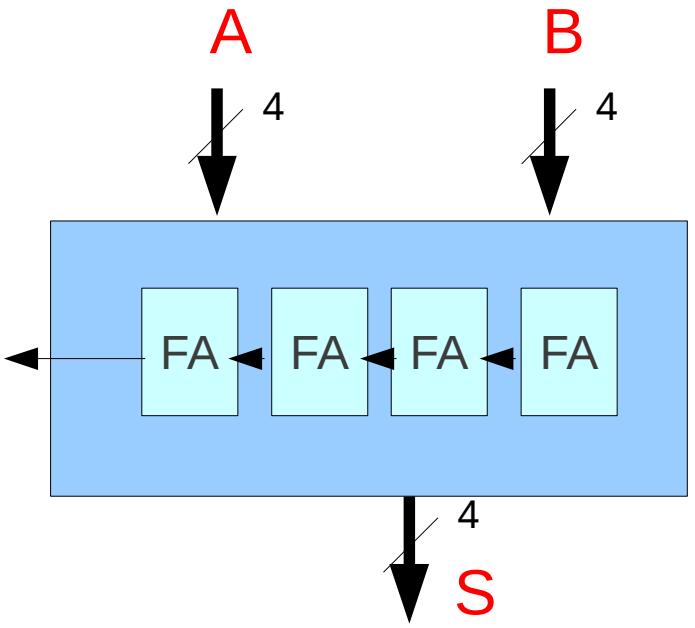
```
A = 3;
B = 7;
C = A + B;
A = 2;
B = 5;
C = A + B;
A = 3;
B = 3;
C = A + B;
```

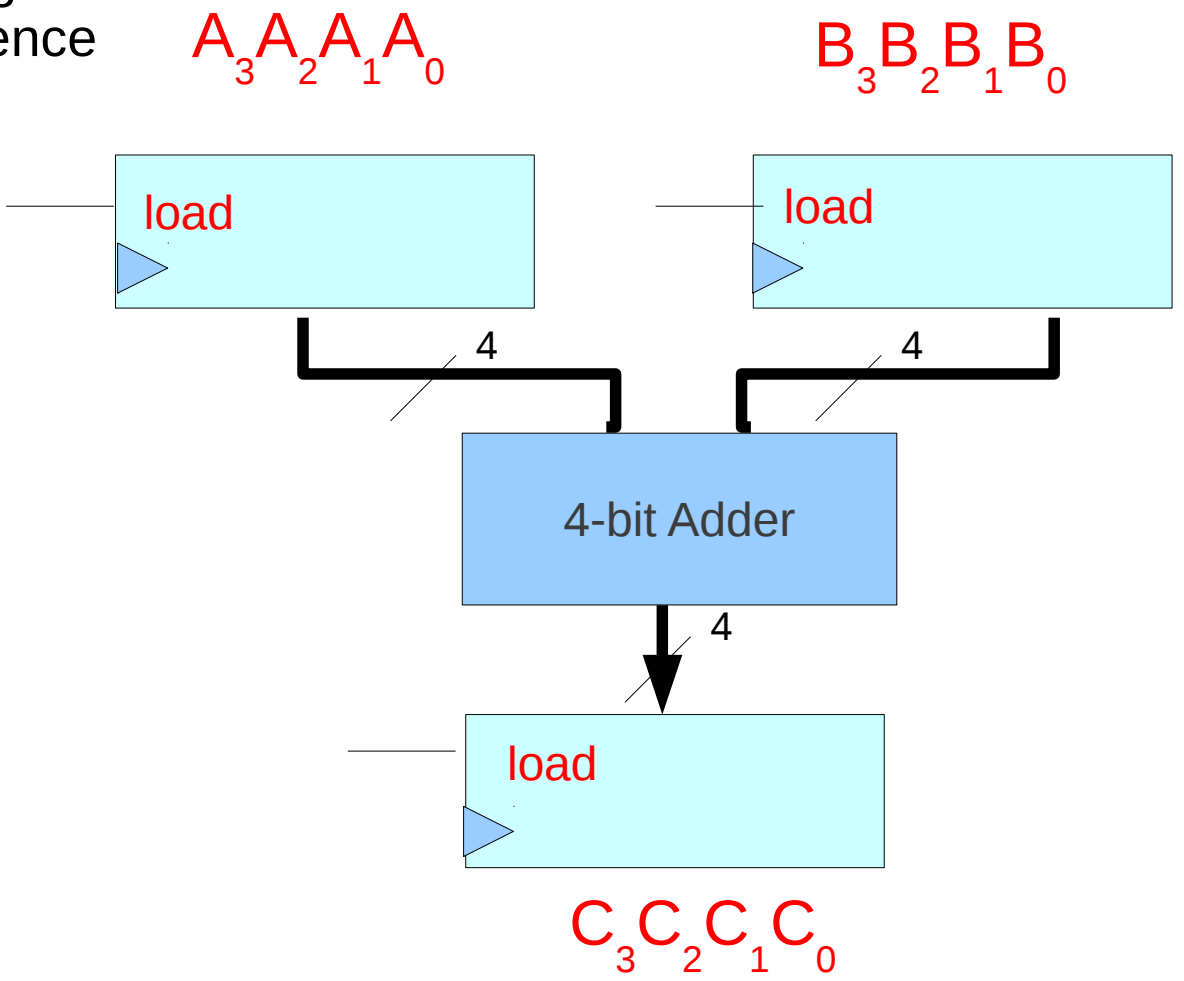
(a) What is a register and why do we need it?
(b) Explain a register with a parallel load.



(c) Explain the hardware design to perform the following sequence of additions

```
A = 3;
B = 7;
C = A + B;
A = 2;
B = 5;
C = A + B;
A = 3;
B = 3;
C = A + B;
```





(d) Draw an example of waveforms using this hardware design

#2 Read the pdf file about the classical implementation of edge triggered FF. Summarize the operations of the FF.