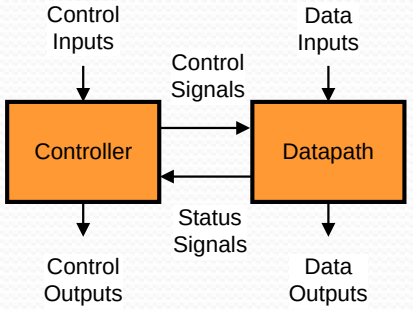
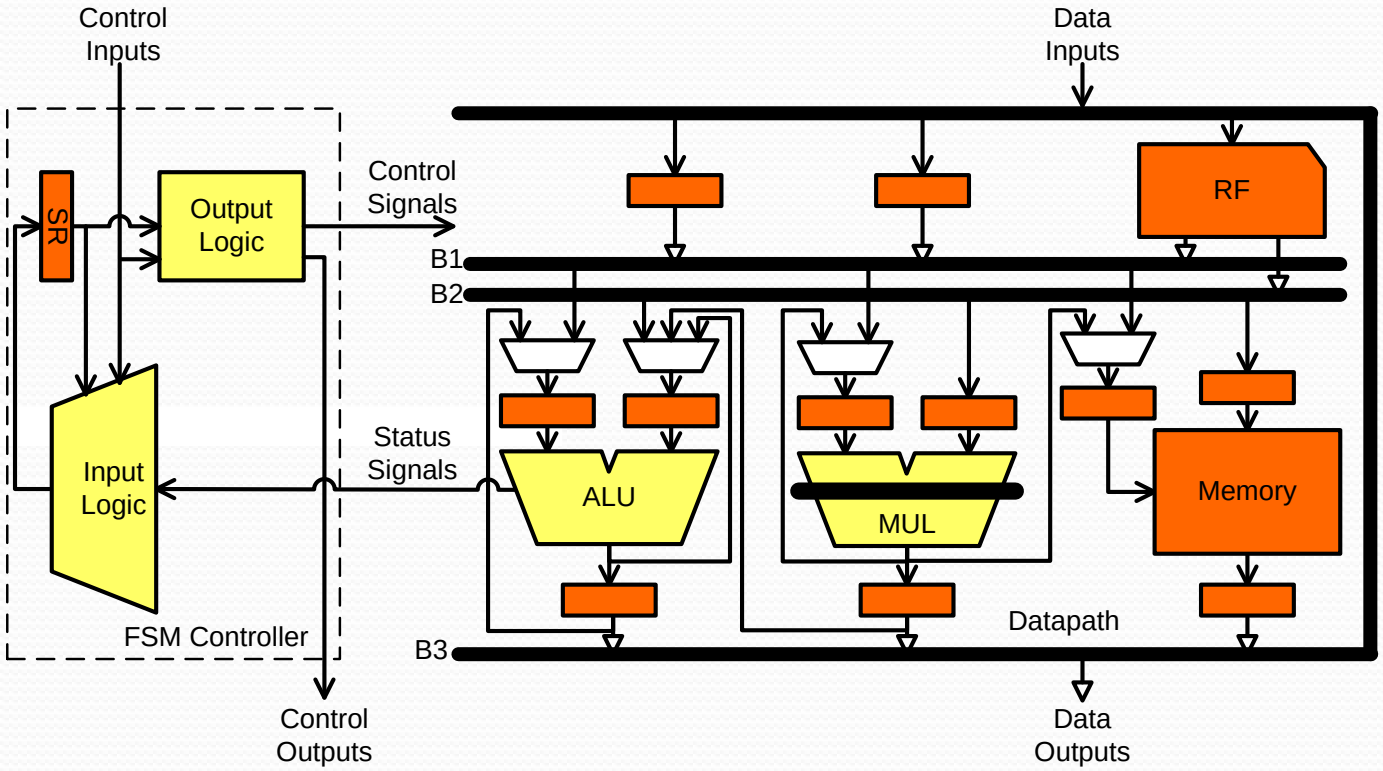
IP Architecture

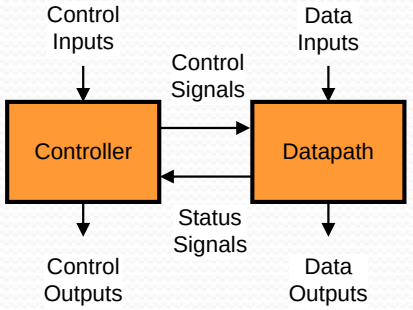
Simple Architecture (Small number of states)

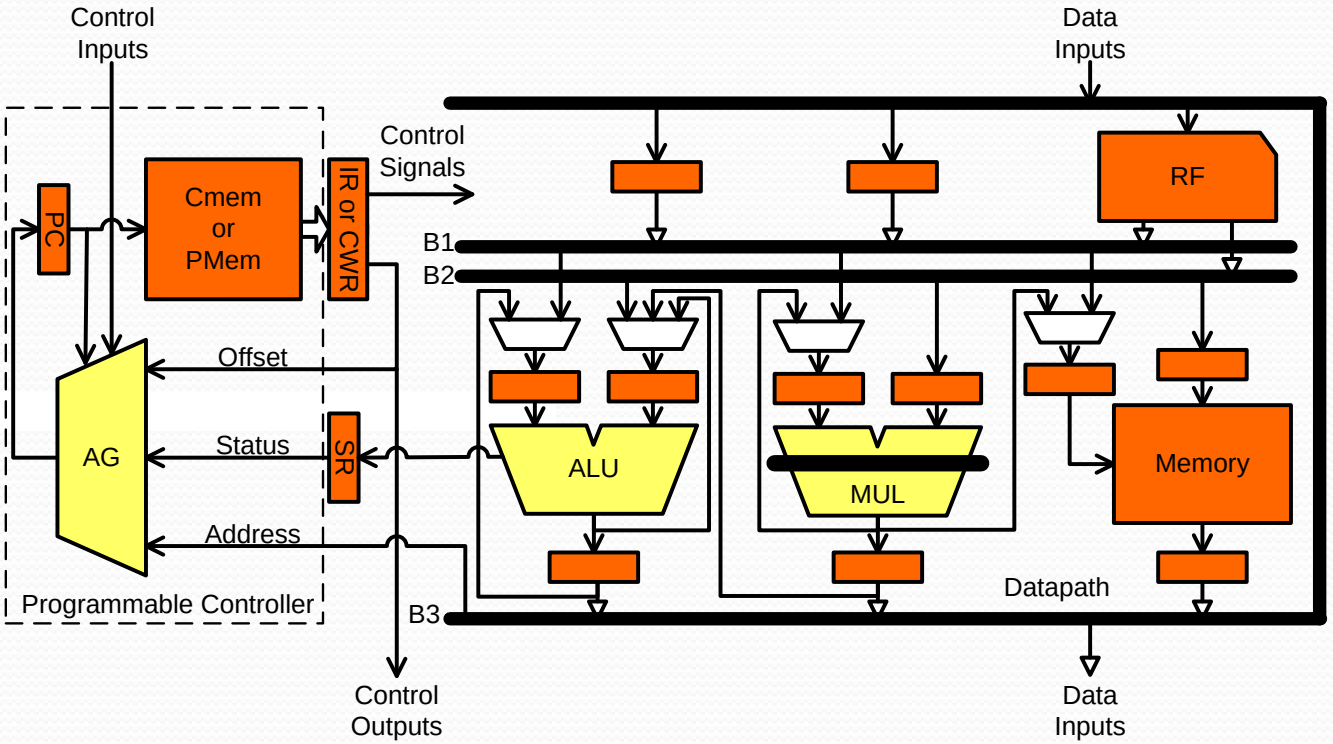




IP Architecture

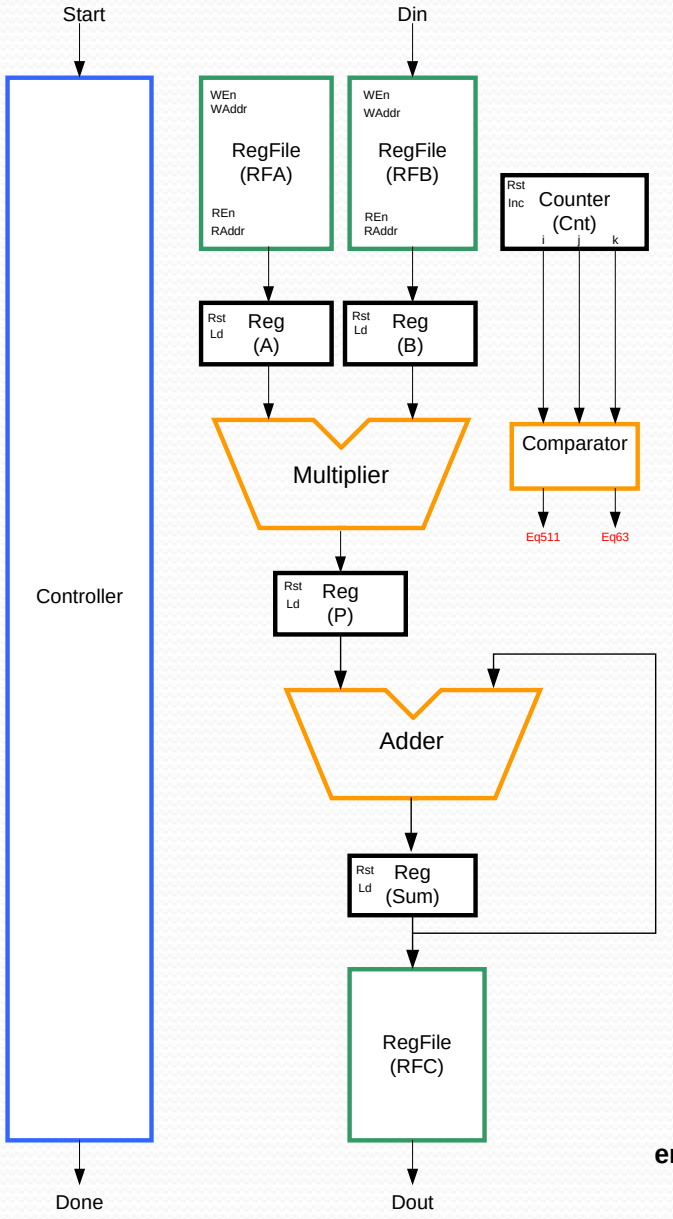
Complex Architecture (Large number of states)





Matrix-Multiply Architecture

```
---------- Matrix Multiplication
for i in 0 to 7 loop
    for j in 0 to 7 loop
        Sum=0;
        for k in 0 to 7 loop
            A := BlockA( i, k );
            B := BlockB( k, j );
            P := A * B;
            Sum := Sum + P;
            if( k = 7 ) then
                BlockC( i, j ) := Sum;
            end if;
        end loop;
    end loop;
end loop;
```



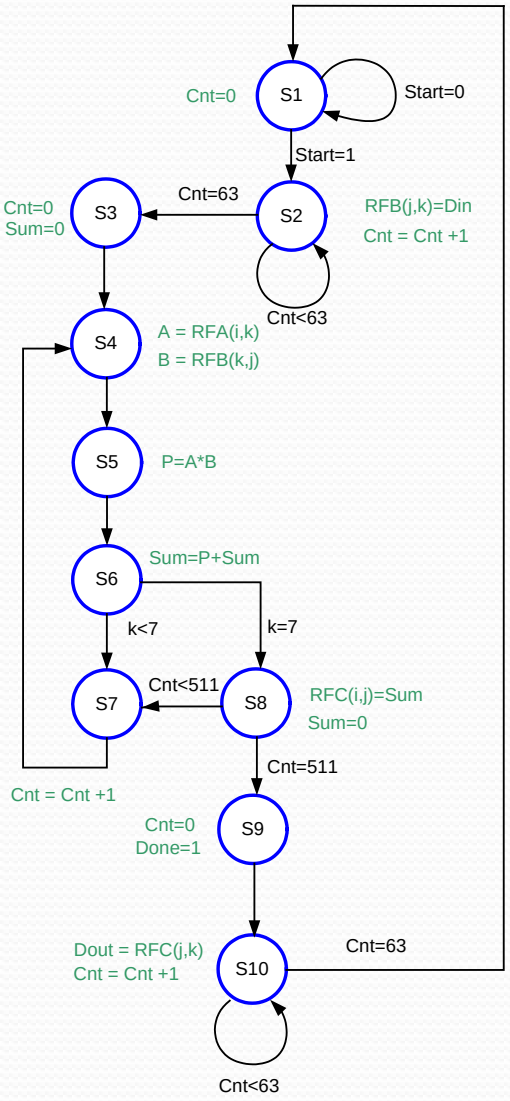

```
---------- Starting
wait until Start = '1';
Done <= '0';

---------- Read Input Data
for i in 0 to 7 loop
    for j in 0 to 7 loop
        wait until Clk = '1' and Clk'event;
        BlockB( i, j ) := Din;
    end loop;
    end loop;

---------- ---Finishing
wait until Clk = '1' and Clk'event;
Done <= '1';

----------Output Data
for i in 0 to 7 loop
    for j in 0 to 7 loop
        wait until Clk = '1' and Clk'event;
        Done <= '0';
        Dout <= BlockC( i, j );
    end loop;
end loop;
end process;
```

Matrix-Multiply FSMD




```
---------- Matrix Multiplication
for i in 0 to 7 loop
    for j in 0 to 7 loop
        Sum=0;
        for k in 0 to 7 loop
            A := BlockA( i, k );
            B := BlockB( k, j );
            P := A * B;
            Sum := Sum + P;
            if( k = 7 ) then
                BlockC( i, j ) := Sum;
            end if;
        end loop;
    end loop;
end loop;
```

```
---------- Starting
wait until Start = '1';
Done <= '0';

---------- Read Input Data
for i in 0 to 7 loop
    for j in 0 to 7 loop
        wait until Clk = '1' and Clk'event;
        BlockB( i, j ) := Din;
    end loop;
    end loop;
```

```
---------- ---Finishing
wait until Clk = '1' and Clk'event;
Done <= '1';

----------Output Data
for i in 0 to 7 loop
    for j in 0 to 7 loop
        wait until Clk = '1' and Clk'event;
        Done <= '0';
        Dout <= BlockC( i, j );
    end loop;
end loop;
end process;
```