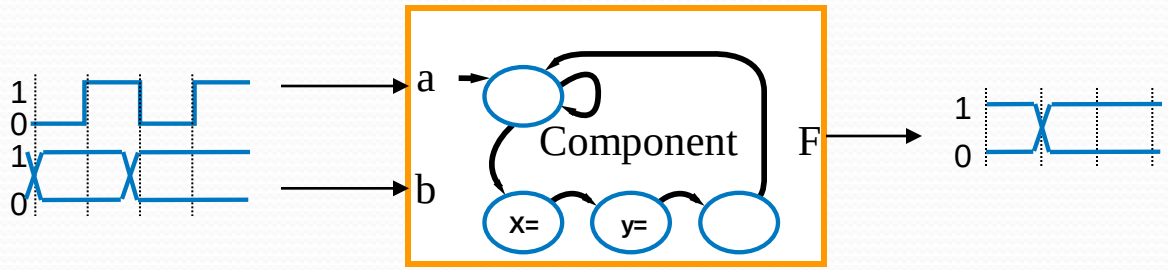
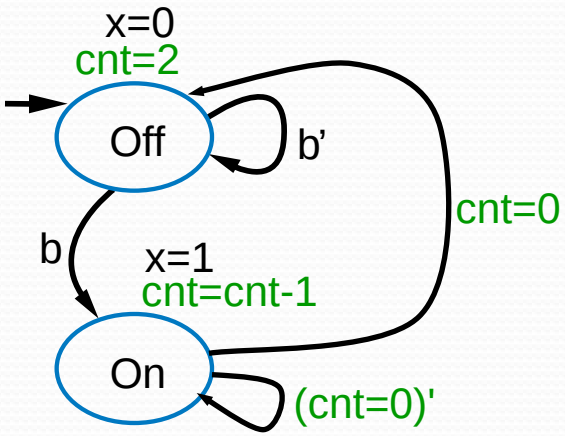
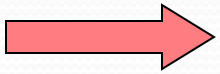
IP Design

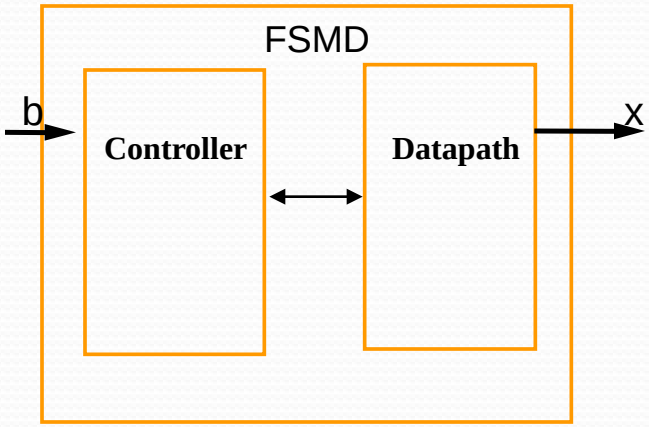



```
void  MatrixMult (int a[][8], int b[][8[], int c[][8]) {
  register int   i, j, k;
  for (i=0; i<8; i++)
    for (j=0; j<8; j++) {
      c[i][j] = 0;
      for (k=0; k<8; k++)
        c[i][j] += a[i][k] * b[k][j];
    }
}
```

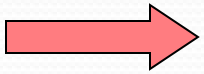
IP Design vs FSMD Design

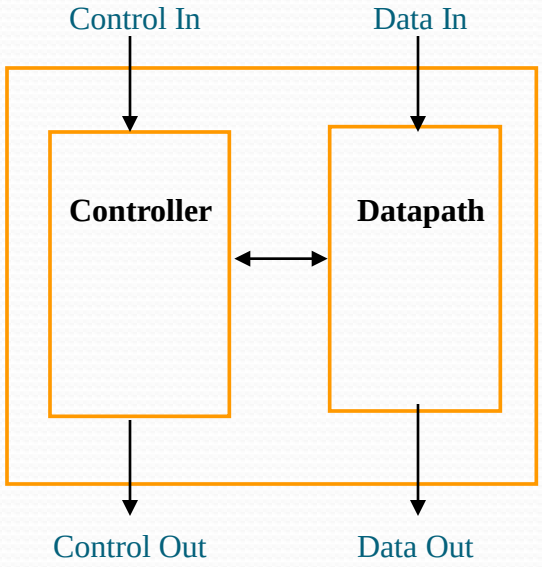







```
void  MatrixMult (int a[][8], int b[][8[, int c[][8]) {
  register int   i, j, k;
  for (i=0; i<8; i++)
    for (j=0; j<8; j++) {
      c[i][j] = 0;
      for (k=0; k<8; k++)
        c[i][j] += a[i][k] * b[k][j];
    }
}
```



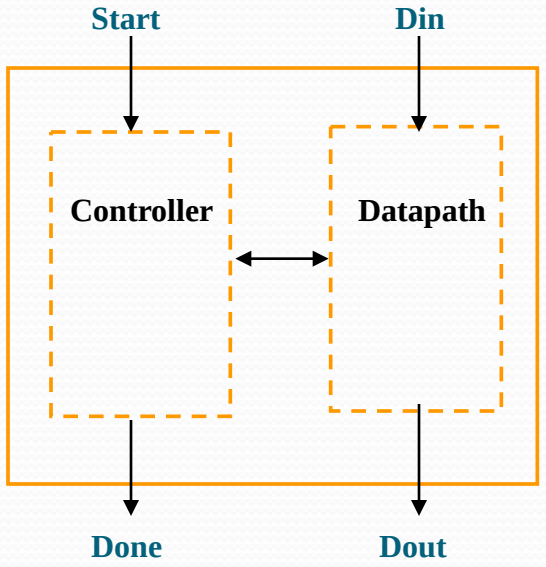


Matrix-Multiply Behavior

```
architecture  MatrixMult_beh of  MatrixMult is
begin
        process
            ...
            ----------------------------- 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;

            ----------------------------- 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;
```




```
            ----------------------------- ---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;

end MatrixMult_beh;
```