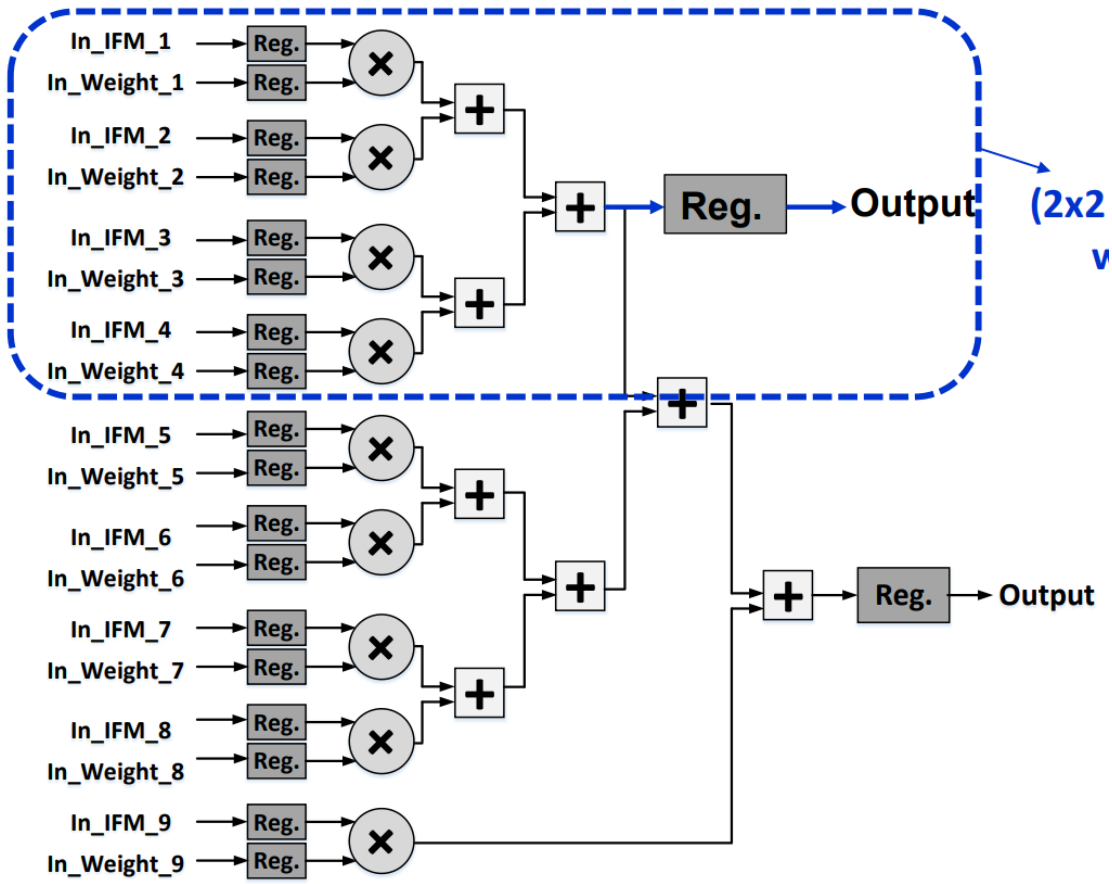
DIC HW3

312510190 張家瑋

(1). Without Pipeline:




```
        Building instance specific data structures.
        Loading native compiled code:     ..................... Done
        Design hierarchy summary:
                                 Instances  Unique
                Modules:                 3       3
                Registers:              37      37
                Scalar wires:            5       -
                Vectored wires:          3       -
                Always blocks:          12      12
                Initial blocks:          5       5
                Simulation timescale:  1ps
        Writing initial simulation snapshot: worklib.TESTBED:v
Loading snapshot worklib.TESTBED:v ..................... Done
SVSEED default: 1
*Verdi* Loading libsscore_ius152.so
ncsim> source /usr/cad/cadence/INCISIV/INCISIVE_15.20.084/tools/inca/files/ncsimrc
ncsim> run
FSDB Dumper for IUS, Release Verdi_P-2019.06, Linux x86_64/64bit, 05/26/2019
(C) 1996 - 2019 by Synopsys, Inc.
*Verdi* FSDB WARNING: The FSDB file already exists. Overwriting the FSDB file may crash the programs that are using this file.
*Verdi* : Create FSDB file 'Convolution_without_pipeline.fsdb'
*Verdi* : Begin traversing the scopes, layer (0).
*Verdi* : Enable +mda dumping.
*Verdi* : End of traversing.
*Verdi* : Begin traversing the scopes, layer (0).
*Verdi* : End of traversing.
PASS PATTERN NO.   0
PASS PATTERN NO.   1
PASS PATTERN NO.   2
PASS PATTERN NO.   3
PASS PATTERN NO.   4
PASS PATTERN NO.   5
```

Without Pipeline				
Cycle Period (ps)	**Area**	**Latency (ps)**	**Throughput (G OP/S)**	**Timing**
1250	34511.9	95625	4.705	MET
1200	32586.6	91800	4.9019	MET
1150	33492.2	87975	5.115	MET
1100	34633.6	84150	5.347	VIOLATE
1050	35000.3	80325	5.602	VIOLATE

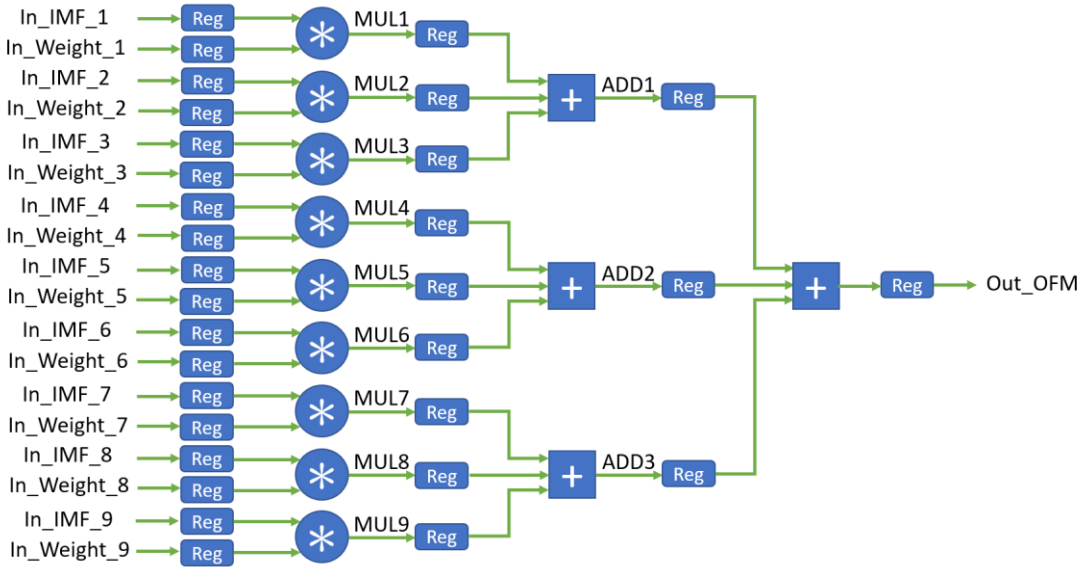
(2). With Pipeline:

```
always @(posedge clk) begin

    if(curr_state == CAL_OUT) begin
        mul_out[0] <= mul_a[0]*Weight_Buffer[0];
        mul_out[1] <= mul_a[1]*Weight_Buffer[1];
        mul_out[2] <= mul_a[2]*Weight_Buffer[2];
        mul_out[3] <= mul_a[3]*Weight_Buffer[3];
        mul_out[4] <= mul_a[4]*Weight_Buffer[4];
        mul_out[5] <= mul_a[5]*Weight_Buffer[5];
        mul_out[6] <= mul_a[6]*Weight_Buffer[6];
        mul_out[7] <= mul_a[7]*Weight_Buffer[7];
        mul_out[8] <= mul_a[8]*Weight_Buffer[8];
    end else begin
        mul_out[0] <= 0;
        mul_out[1] <= 0;
        mul_out[2] <= 0;
        mul_out[3] <= 0;
        mul_out[4] <= 0;
        mul_out[5] <= 0;
        mul_out[6] <= 0;
        mul_out[7] <= 0;
        mul_out[8] <= 0;
    end
end
```

With Pipeline				
Cycle Period (ps)	**Area**	**Latency (ps)**	**Throughput (G op/s)**	**Timing**
1150	35736.1	89125	5.049	MET
1000	35076.4	77500	5.806	MET
900	36406.6	69750	6.451	MET
850	37371.4	65875	6.831	MET
800	37879	62000	7.258	MET
700	38180	54250	8.189	VIOLATE

(3)Optimistic



	OPTIMISTIC				
Cycle Period (ps)	**Area**	**Latency (ps)**	**Throughput (G op/s)**	**Area Efficient (Gops/mm)**	**Timing**
850	36072.7	45475	9.895	274.3	MET
700	36097.9	37450	12.016	332.8	MET
600	37057.2	32100	14.018	378.2	MET
550	37796.9	29425	15.293	404.6	MET
500	39154.6	39750	16.822	429.6	VIOLATE