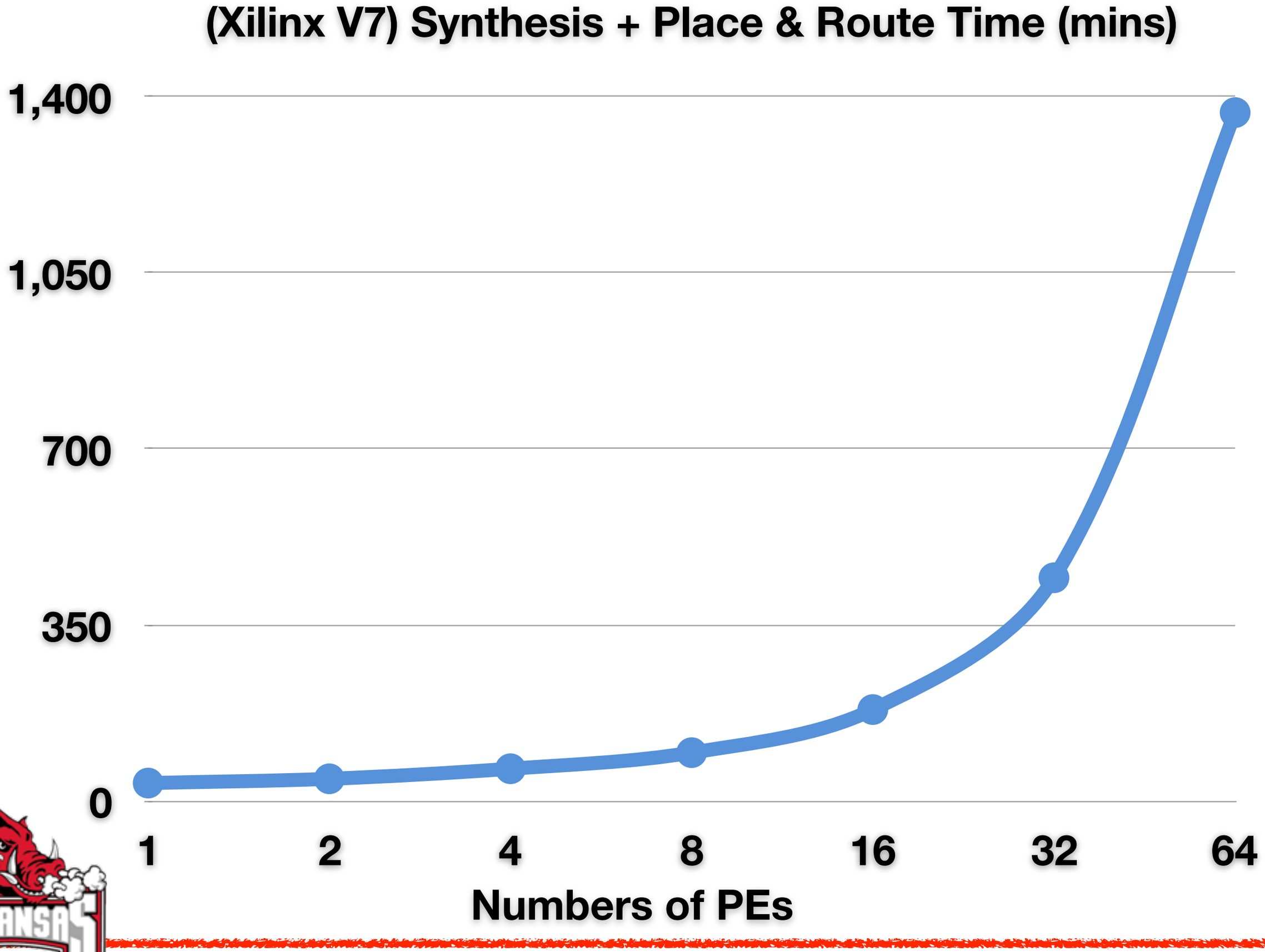
# Just In Time Assembly of Accelerators

**Sen Ma, Zeyad Aklah, David Andrews**
**University of Arkansas**
**2016.02**

# Do you enjoy waiting for synthesis?


(Xilinx V7) Synthesis + Place & Route Time (mins)
1,400
1,050
700
350
0
1
2
4
8
16
32
64
Numbers of PEs


(Xilinx V7) Synthesis + Place & Route Time (mins)
1,400
1,050
700
350
0
1
2
4
8
16
32
64
Numbers of PEs

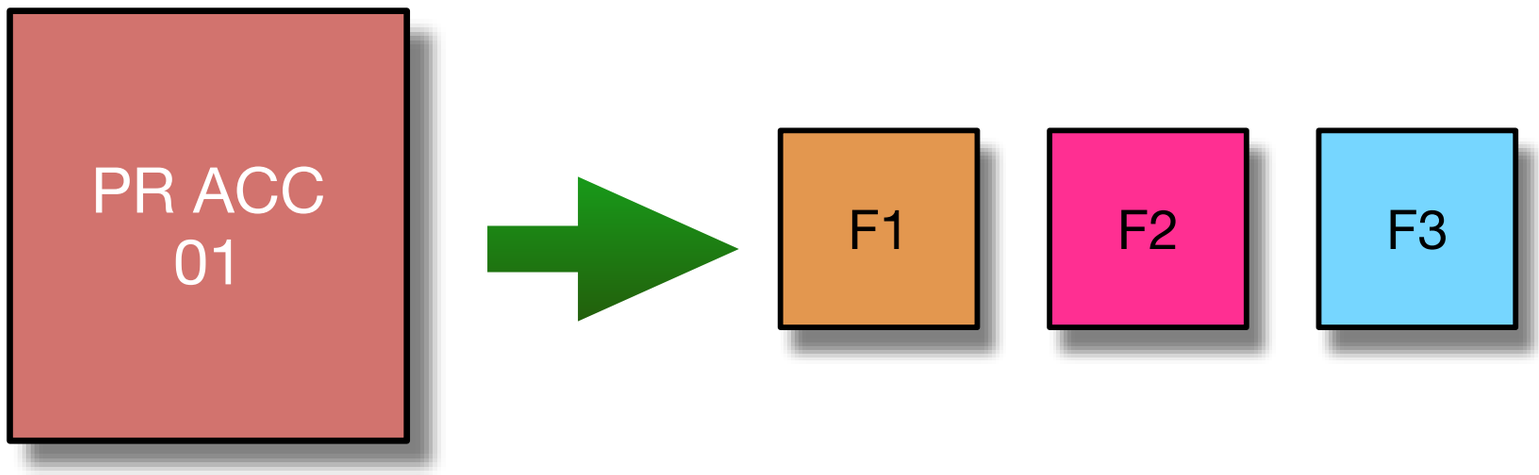
```
S=AVG (dA,SA)  ;

S=VMUL(dA  ,  dB,S);

S=RED   (dB,SB) ;
```

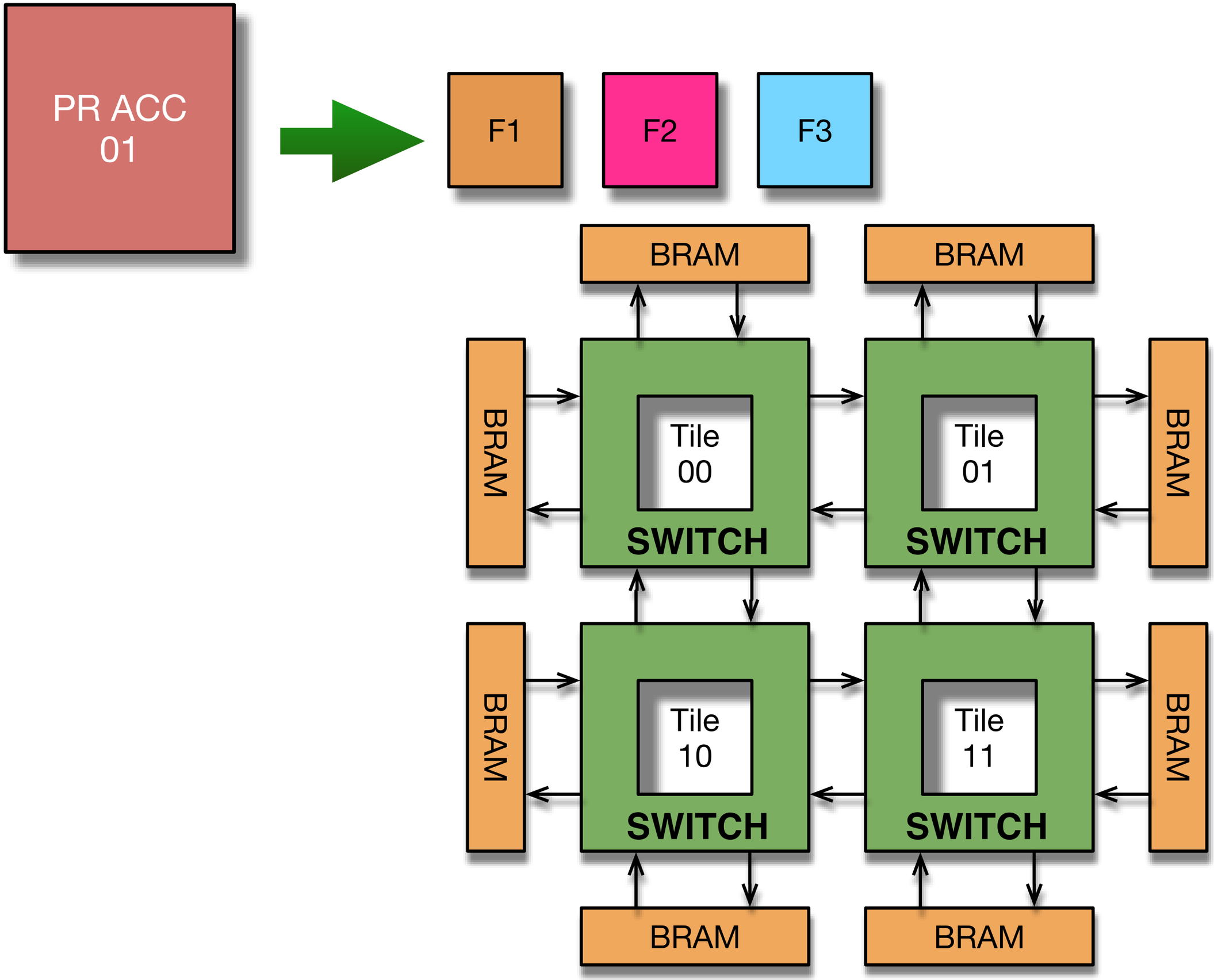
***S=VMUL( AVG (dA,SA) , RED (dB,SB) , S);***

***S=VMUL( RED(dA,SA), AVG (dB,SB) , S);***

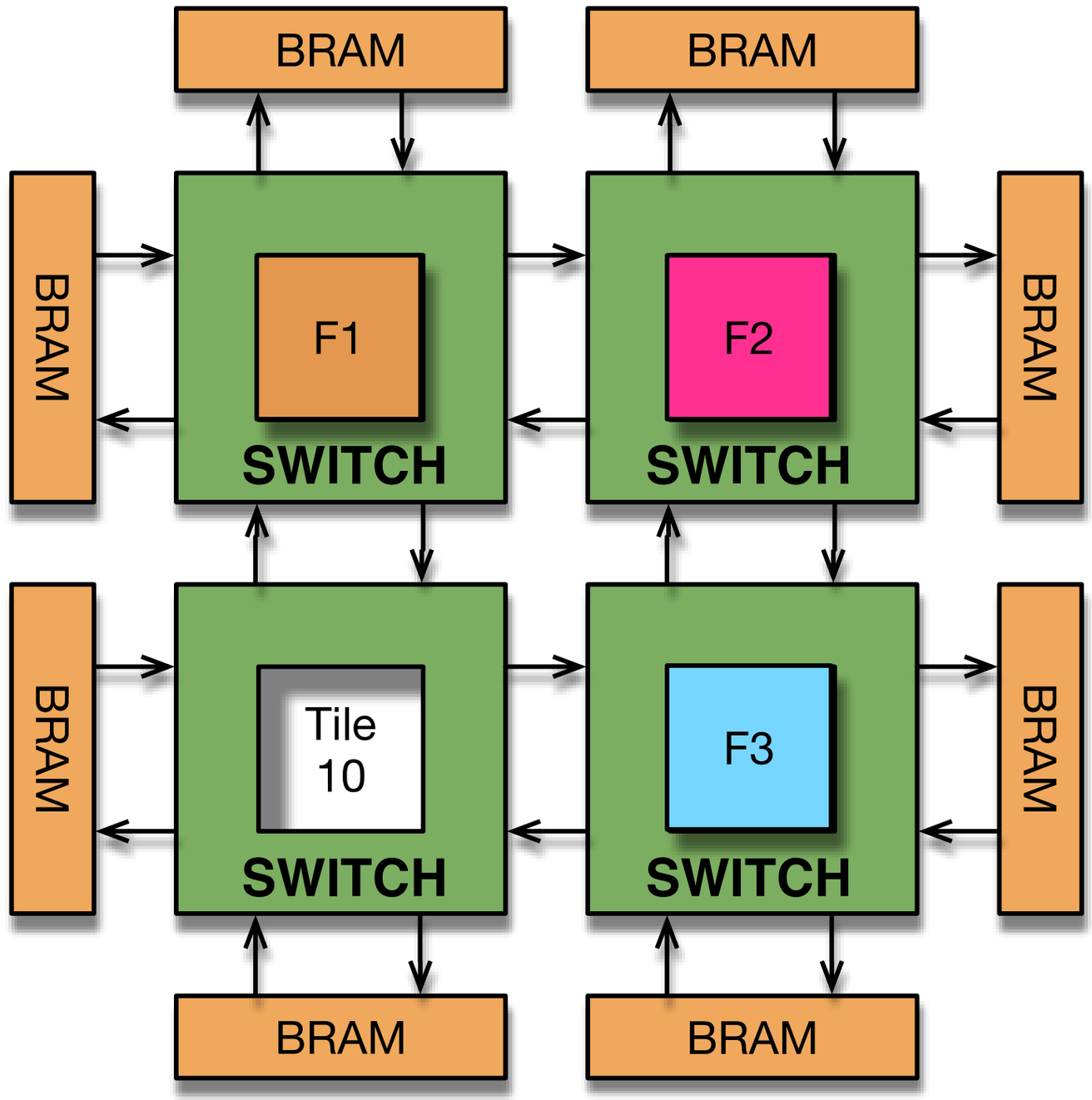
$$S = f_3( f_1 (dA, SA) , f_2 (dB, SB) , S);$$

$$S = f_3( f_1 (dA, SA) , f_2 (dB, SB) , S);$$

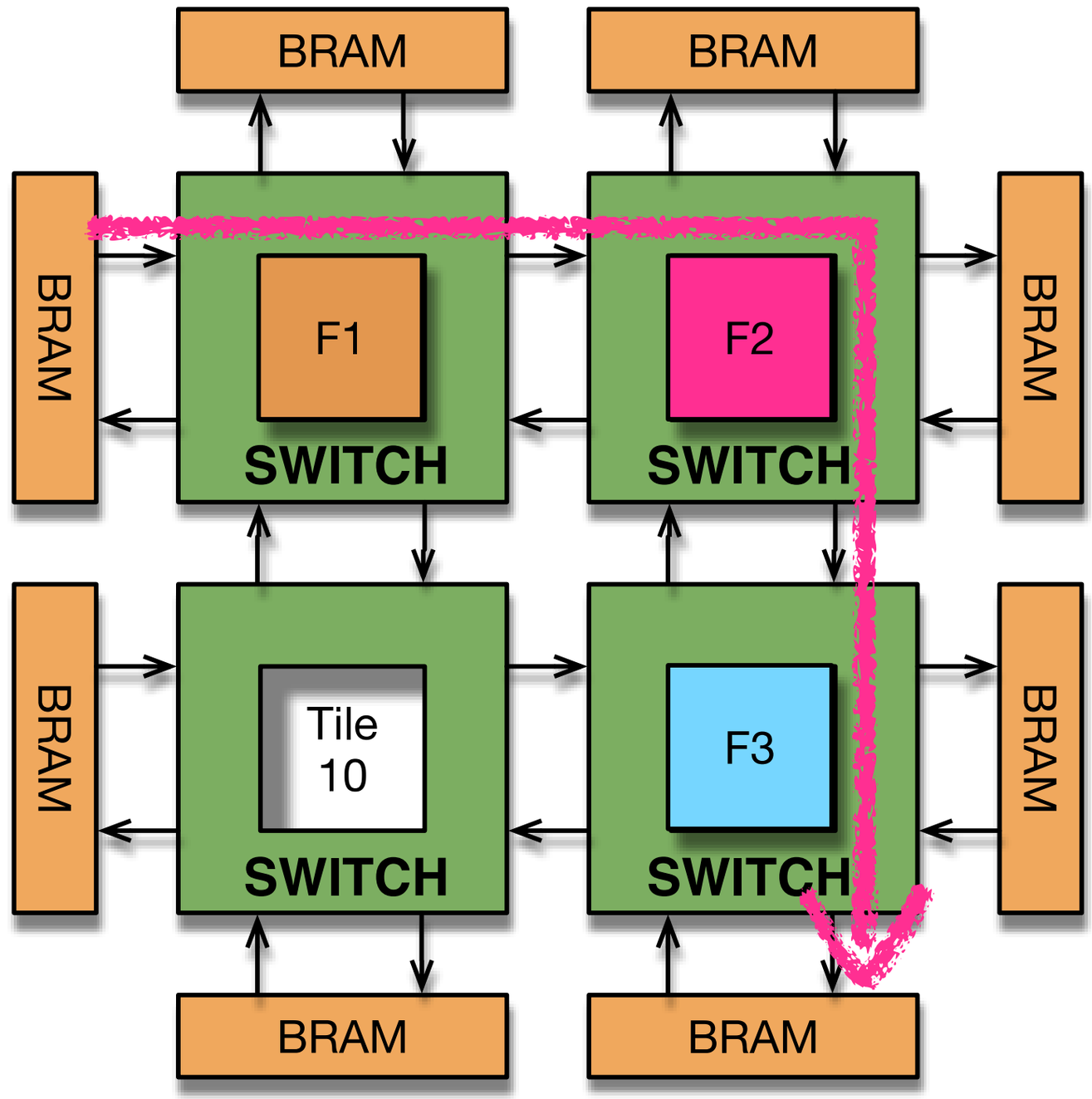


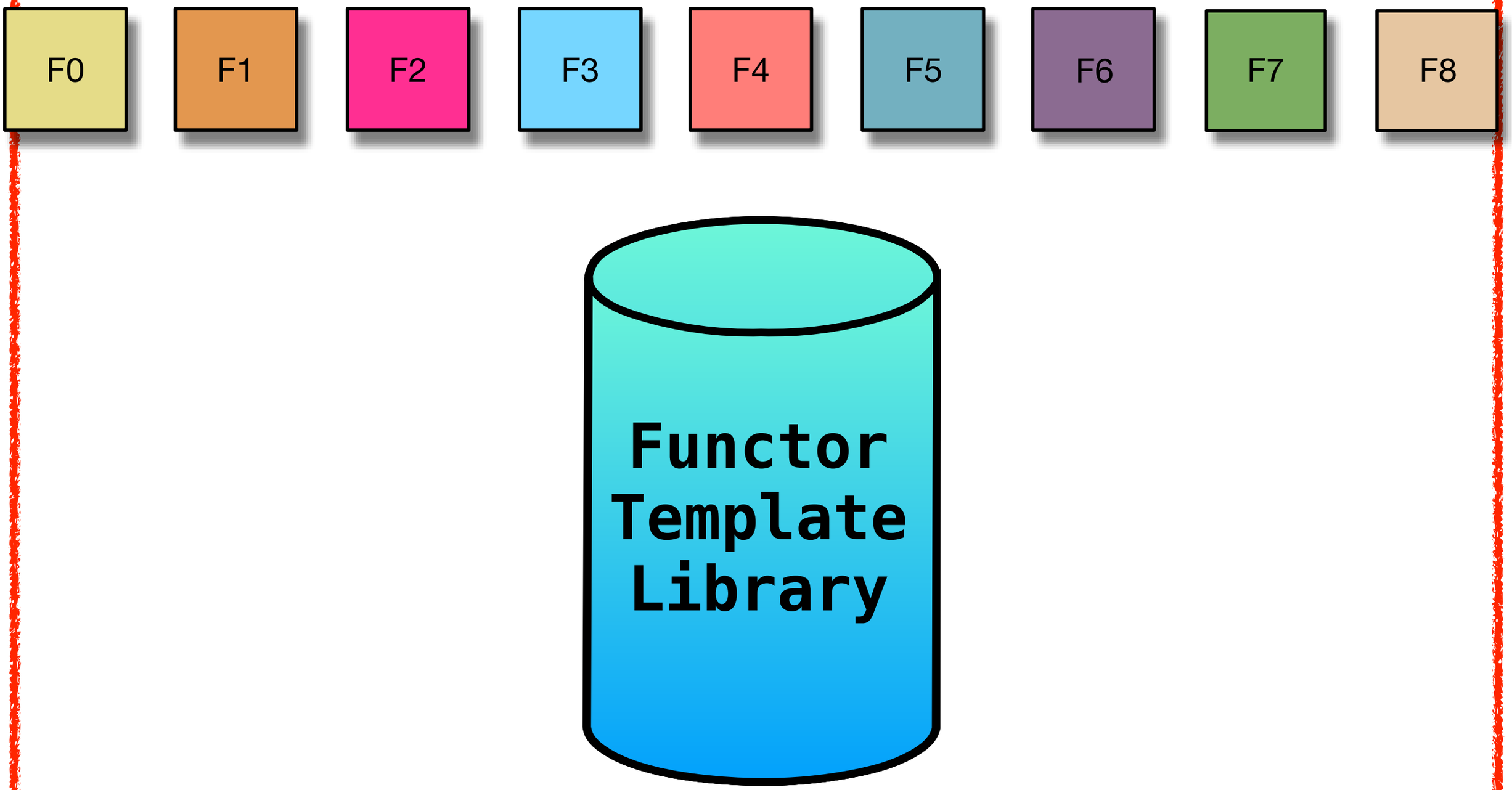

PR ACC
01
F1
F2
F3
BRAM
BRAM
BRAM
BRAM
Tile
00
Tile
01
SWITCH
SWITCH
BRAM
BRAM
Tile
10
Tile
11
SWITCH
SWITCH
BRAM
BRAM

PR ACC
01


BRAM
BRAM
BRAM
F1
F2
BRAM
SWITCH
SWITCH
Tile
10
F3
BRAM
SWITCH
SWITCH
BRAM
BRAM

PR ACC
01


BRAM
BRAM
BRAM
BRAM
F1
F2
SWITCH
SWITCH
Tile
10
F3
SWITCH
SWITCH
BRAM
BRAM


F0
F1
F2
F3
F4
F5
F6
F7
F8
Functor
Template
Library

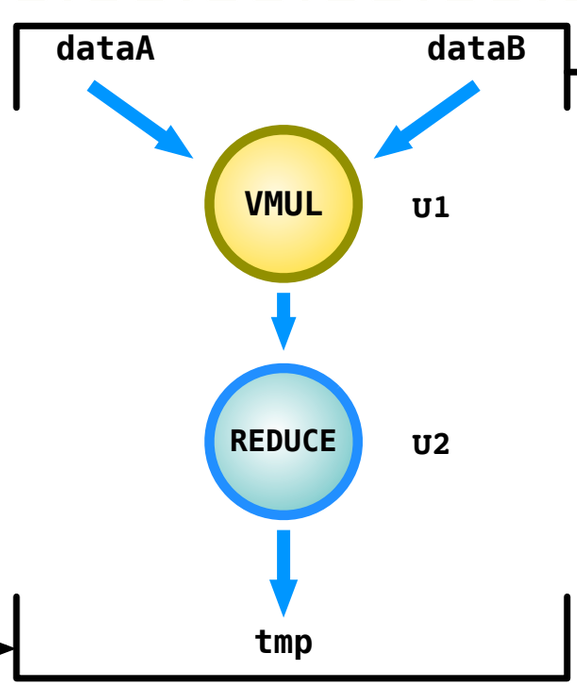
## User Application

```
int tmp = 0;
tmp = REDUCE (
    VMUL (dataA, dataB, Size) );
```

Compile DSL

Build IR

Extract Patterns




**Interpreter Call Generation**

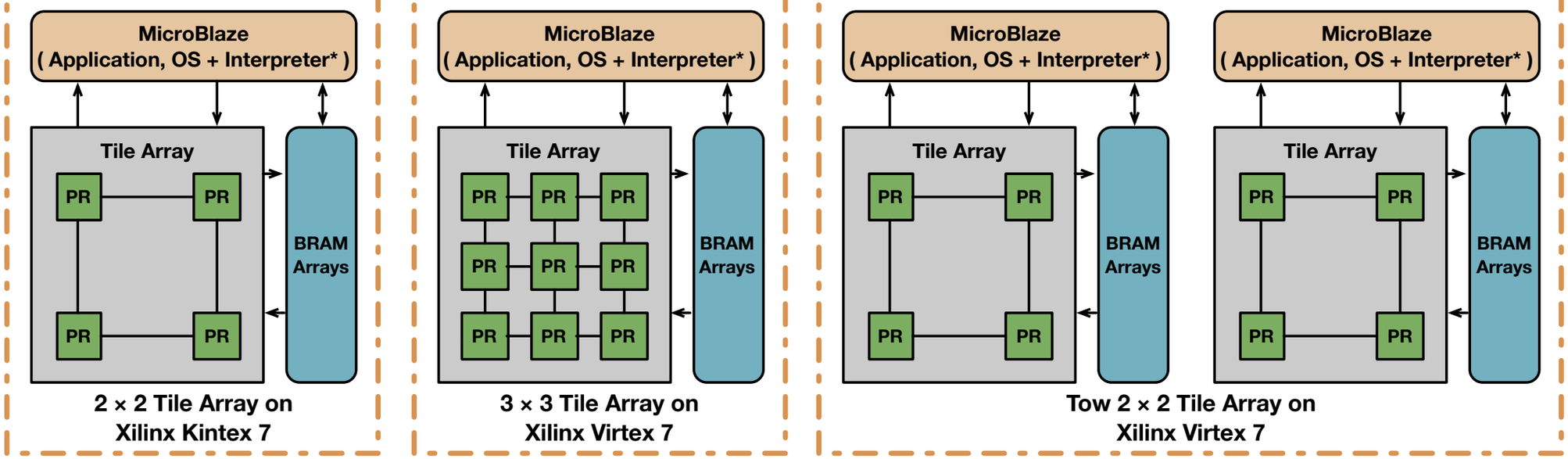
```
VAM_GET_Tile &PR1 U1
VAM_LOAD_Tile  PR1 VMUL
VAM_GET_BRAM &BRAM1 dataA
VAM_GET_Tile &PR2 U2
VAM_GET_BRAM &BRAM2 dataB
VAM_LOAD_Tile  PR2 REDUCE
VAM_GET_BRAM &BRAM3 tmp
VAM_DMA dataA BRAM1
VAM_DMA dataB BRAM2
VAM_ROUTE PR1 PR2
VAM_START PR1 PR2
VAM_DONE PR1 PR2
VAM_DMA BRAM3 tmp
```

*Compiler*

BitStreams Repository

Portable Calls

Bit-Streams




* Interpreters can be implemented in SW or in HW.

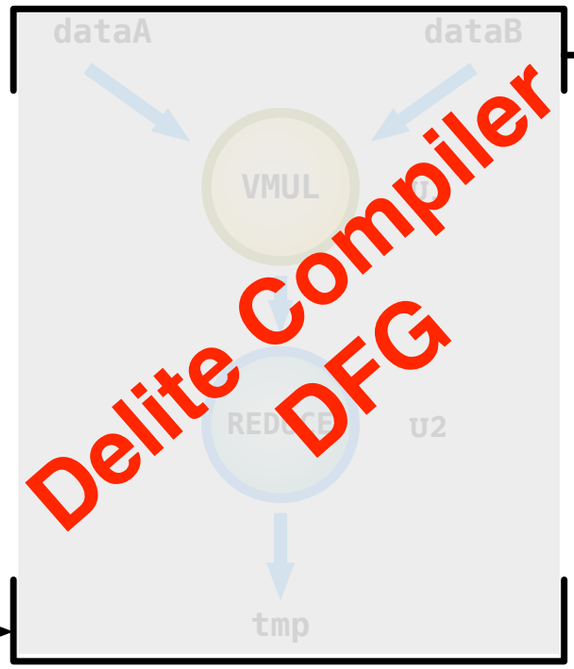
## User Application

```
int tmp = 0;
tmp = REDUCE (
    VMUL (dataA, dataB, Size) );
```

Compile DSL

Build IR

Extract Patterns




**Interpreter Call Generation**

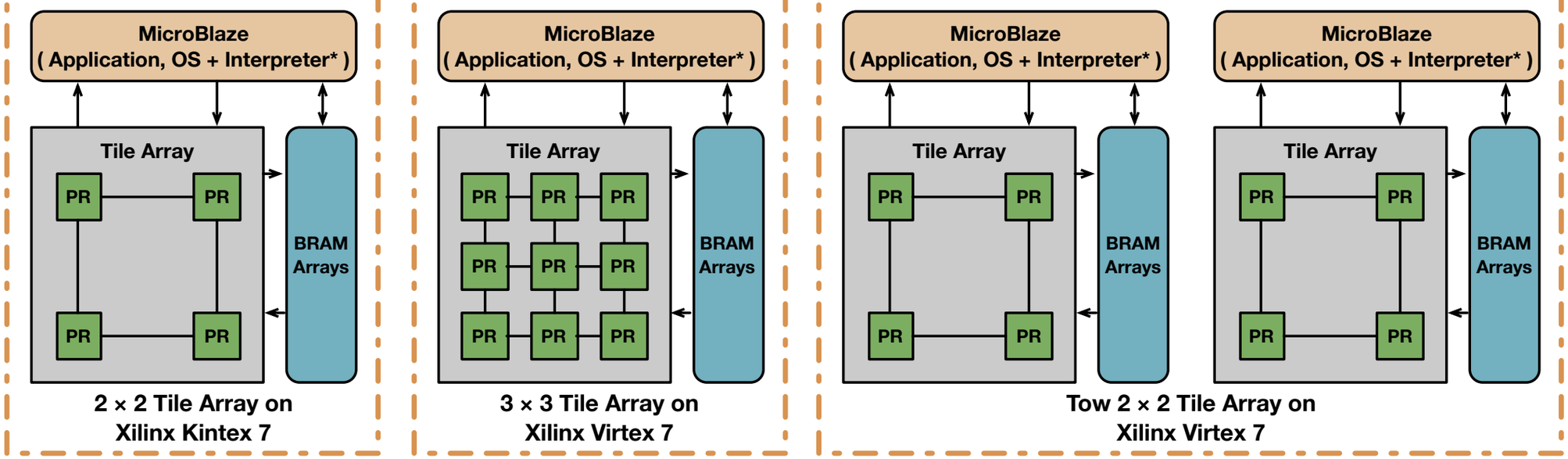
```
VAM_GET_Tile &PR1 U1
VAM_LOAD_Tile  PR1 VMUL
VAM_GET_BRAM &BRAM1 dataA
VAM_GET_Tile &PR2 U2
VAM_GET_BRAM &BRAM2 dataB
VAM_LOAD_Tile  PR2 REDUCE
VAM_GET_BRAM &BRAM3 tmp
VAM_DMA dataA BRAM1
VAM_DMA dataB BRAM2
VAM_ROUTE PR1 PR2
VAM_START PR1 PR2
VAM_DONE PR1 PR2
VAM_DMA BRAM3 tmp
```

*Compiler*

BitStreams Repository

Portable Calls

Bit-Streams




* Interpreters can be implemented in SW or in HW.

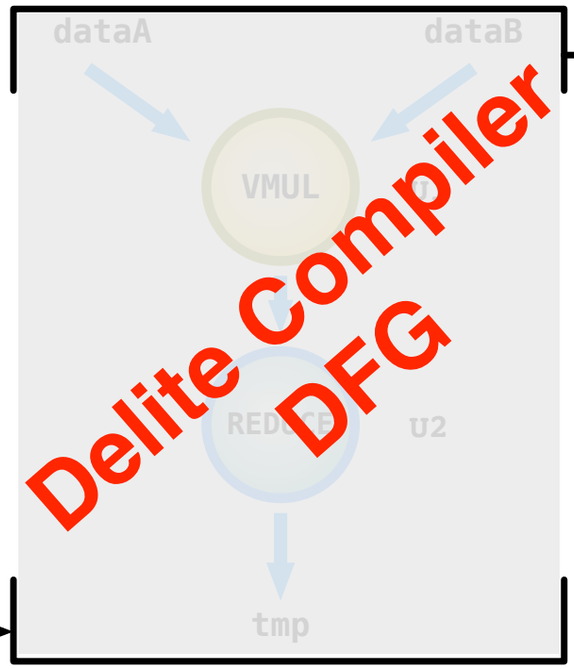
## User Application

```
int tmp = 0;
tmp = REDUCE (
    VMUL (dataA, dataB, Size) );
```

Compile DSL

Build IR

Extract Patterns




Interpreter Call Generation

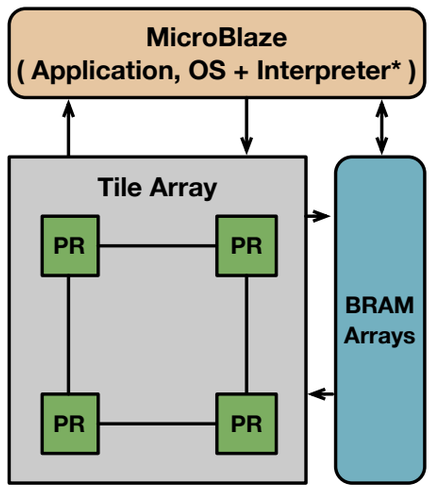
**DFG Executable is Equivalent of Java Byte Code**

*Compiler*

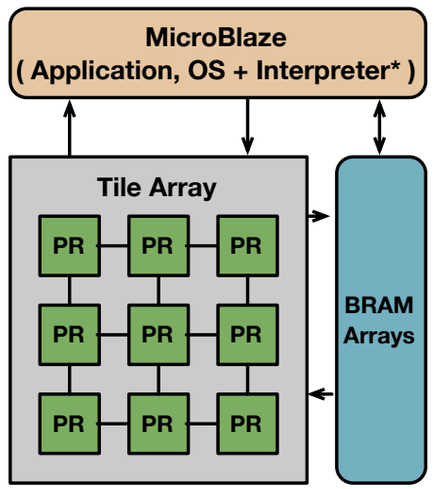
BitStreams Repository

Portable Calls

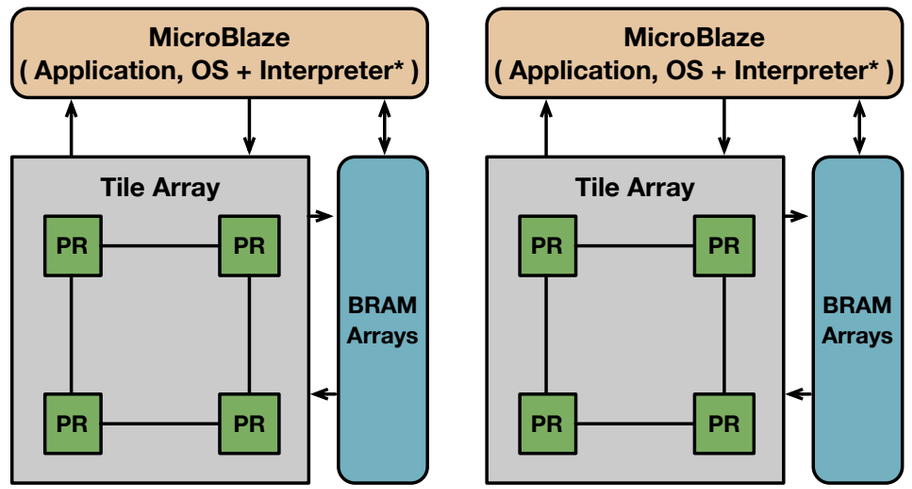
Bit-Streams




2 × 2 Tile Array on Xilinx Kintex 7




3 × 3 Tile Array on Xilinx Virtex 7




Tow 2 × 2 Tile Array on Xilinx Virtex 7

* Interpreters can be implemented in SW or in HW.

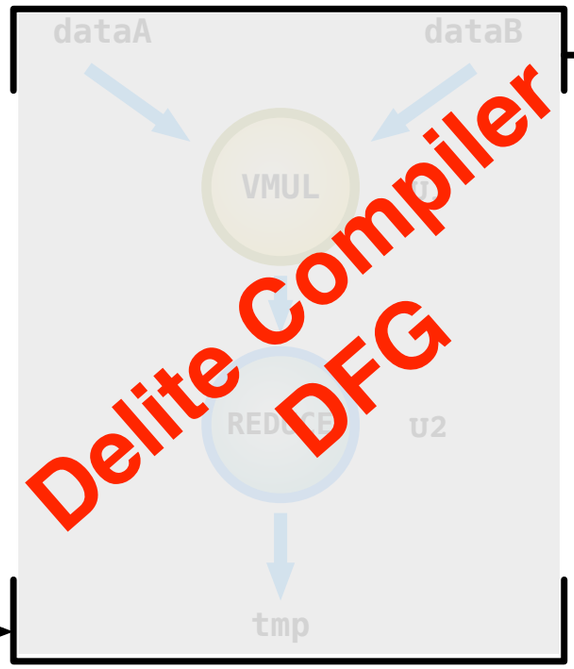
## User Application

```
int tmp = 0;
tmp = REDUCE (
    VMUL (dataA, dataB, Size) );
```

Compile DSL

Build IR

Extract Patterns

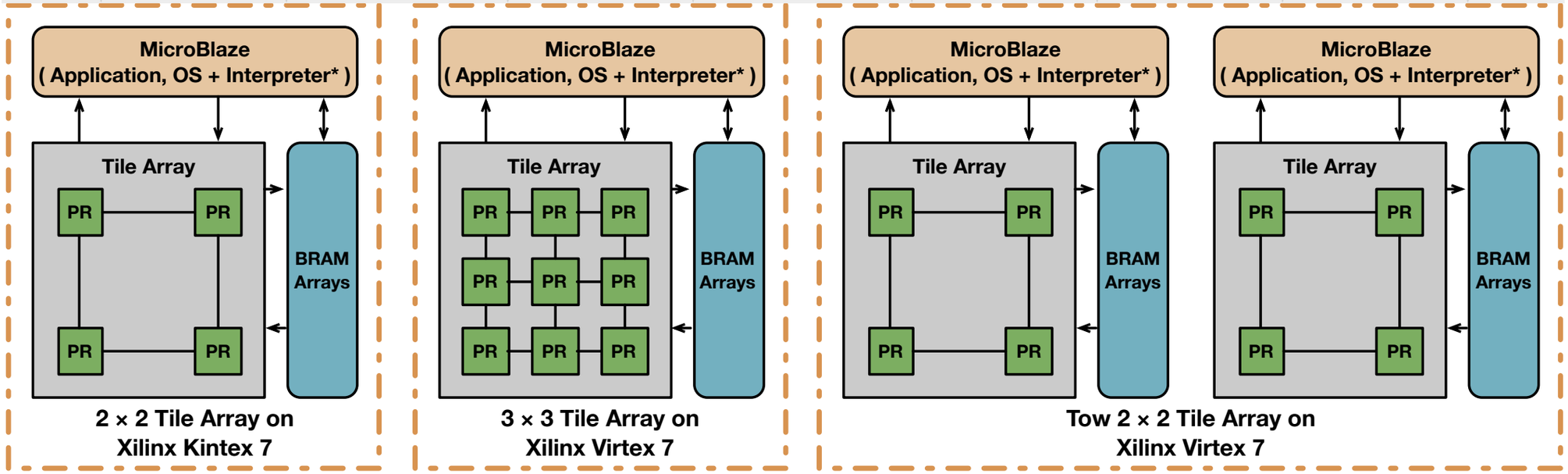



Interpreter Call Generation

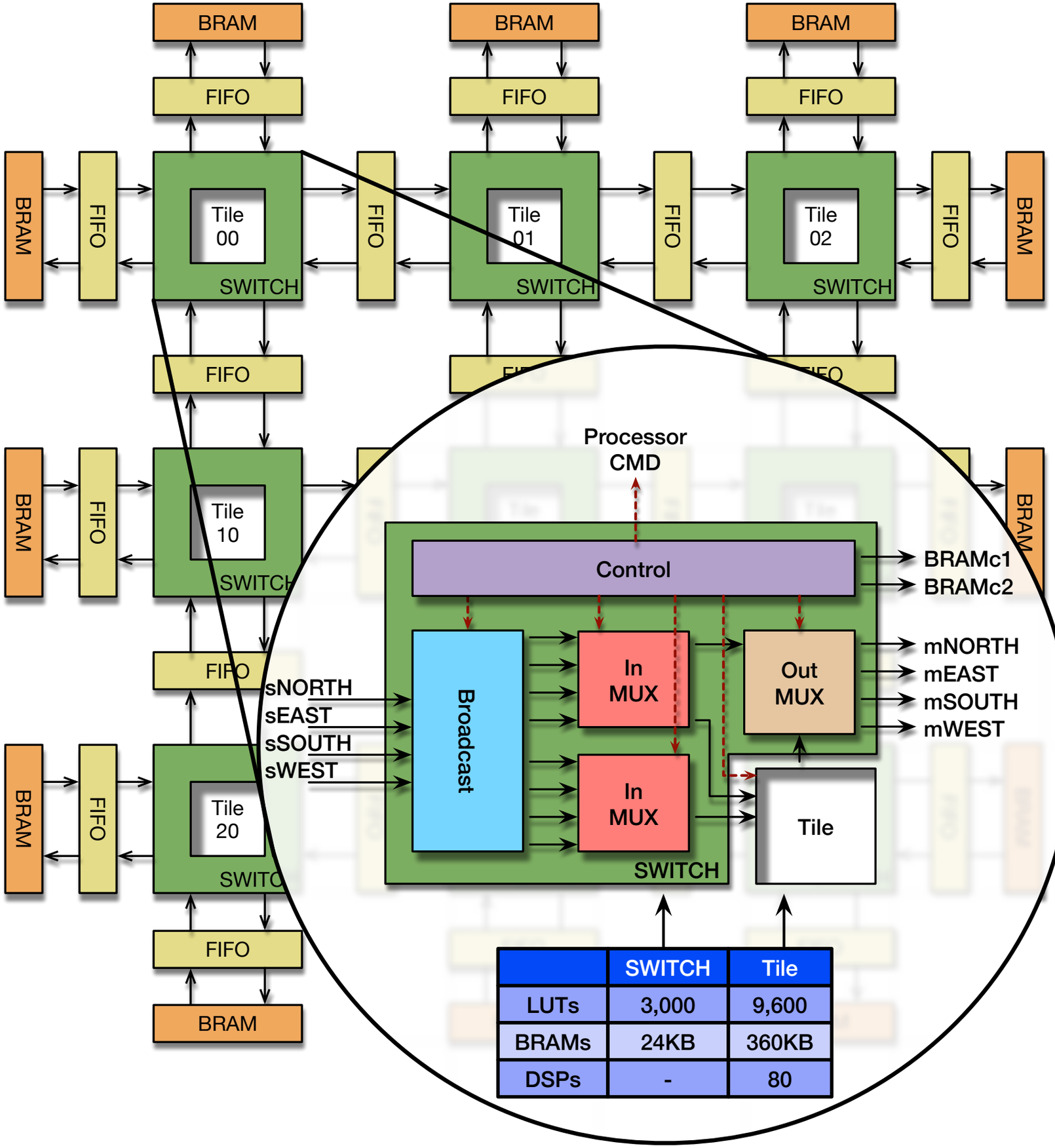
**DFG Executable is Equivalent of Java Byte Code**

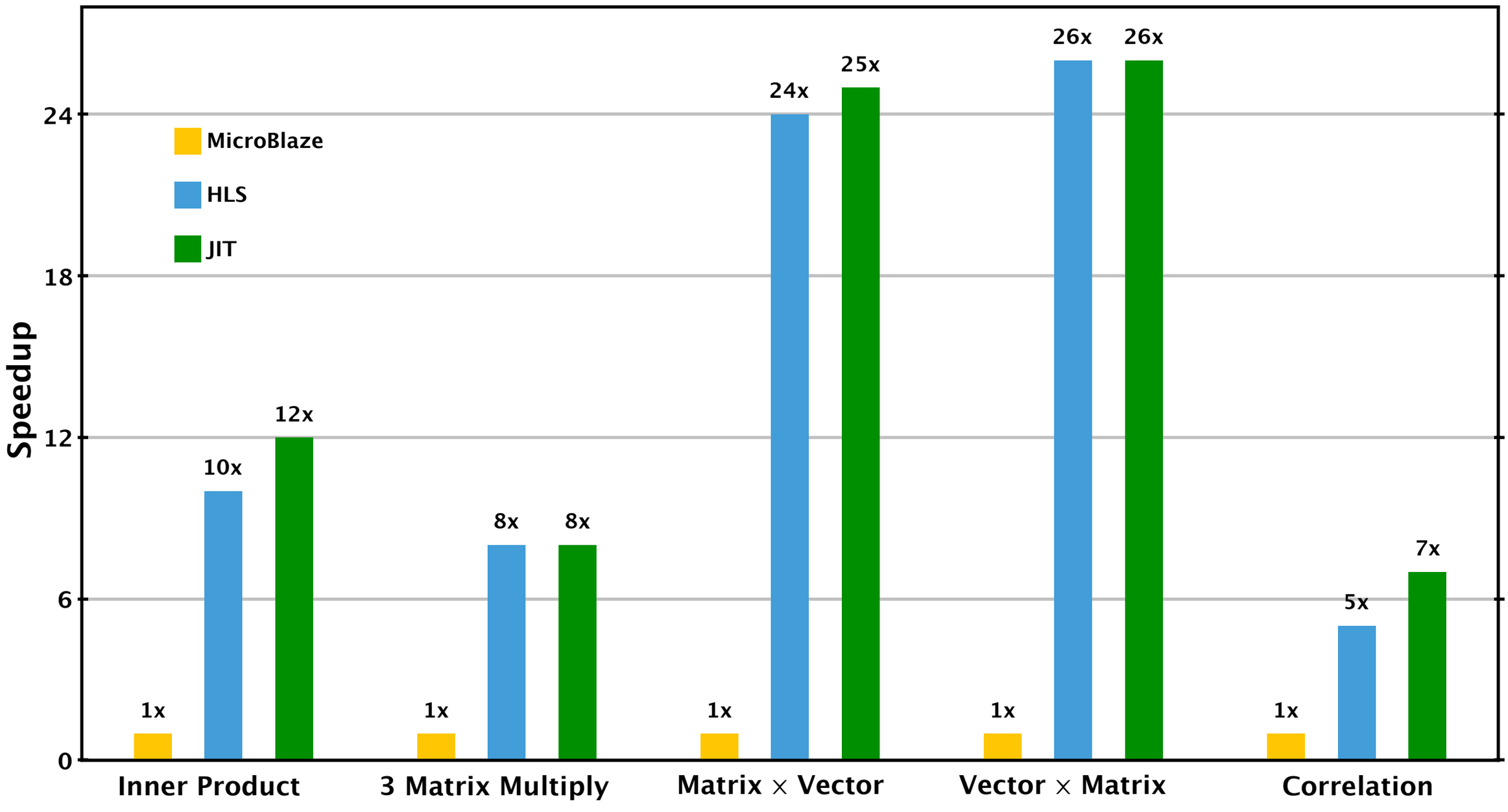
*Compiler*

# Interpreter (aka JVM) allows DFG to execute on All platforms : Enables Portability



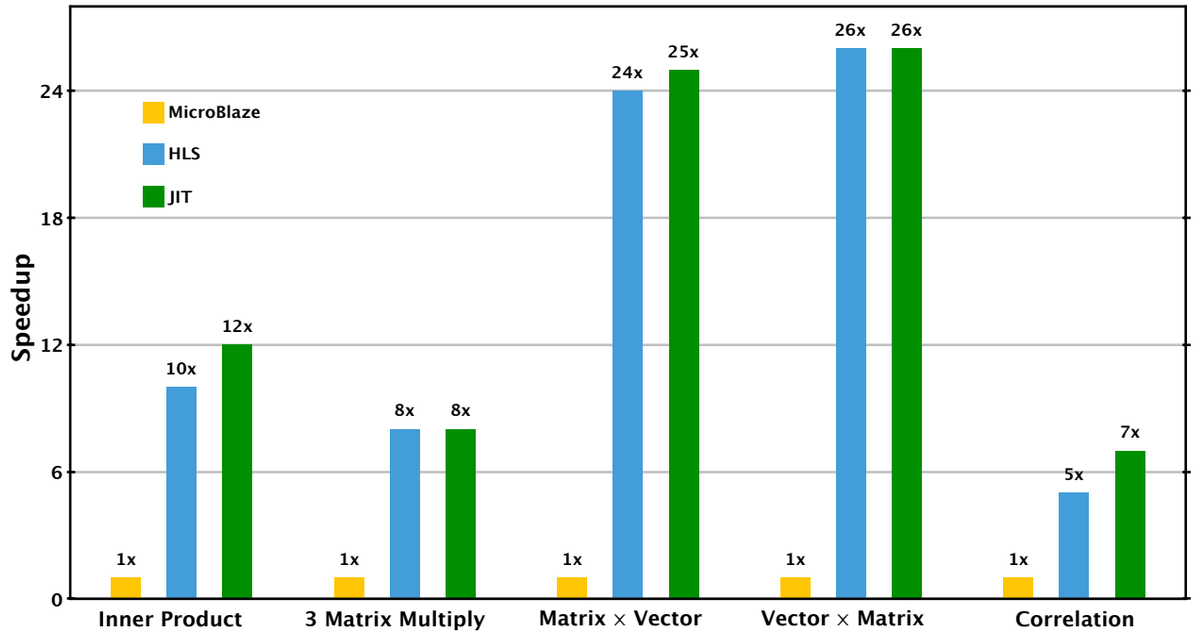

* Interpreters can be implemented in SW or in HW.


BRAM
FIFO
Tile
00
Tile
01
Tile
02
Tile
10
Tile
20
SWITCH
Processor
CMD
Control
BRAMc1
BRAMc2
mNORTH
mEAST
mSOUTH
mWEST
sNORTH
sEAST
sSOUTH
sWEST
Broadcast
In
MUX
Out
MUX
Tile
SWITCH
SWITCH
Tile
LUTs
3,000
9,600
BRAMs
24KB
360KB
DSPs
-
80


Speedup
0
6
12
18
24
MicroBlaze
HLS
JIT
1x
10x
12x
1x
8x
8x
1x
24x
25x
1x
26x
26x
1x
5x
7x
Inner Product
3 Matrix Multiply
Matrix × Vector
Vector × Matrix
Correlation

# Interesting:

- Size of overlay tile.
- Topology.
- Interconnect overhead.
- Programming model.



# Thank you!
# Come to see us in poster session!