CGRA COOKBOOK 2018

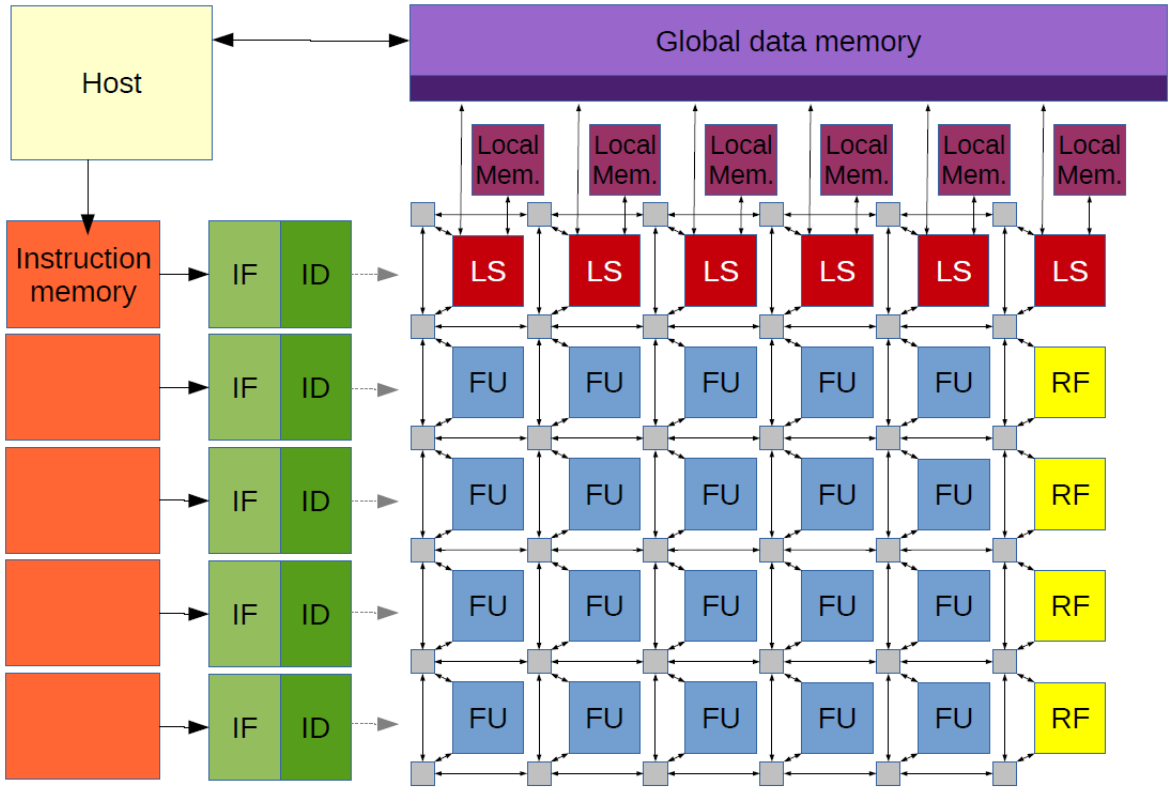
Task

Implementing Edge Detection on the CGRA

Task

Implementing Edge Detection on the CGRA

Naive implementation of Edge Detection



Goal

Finding the best performance for minimal energy & area.

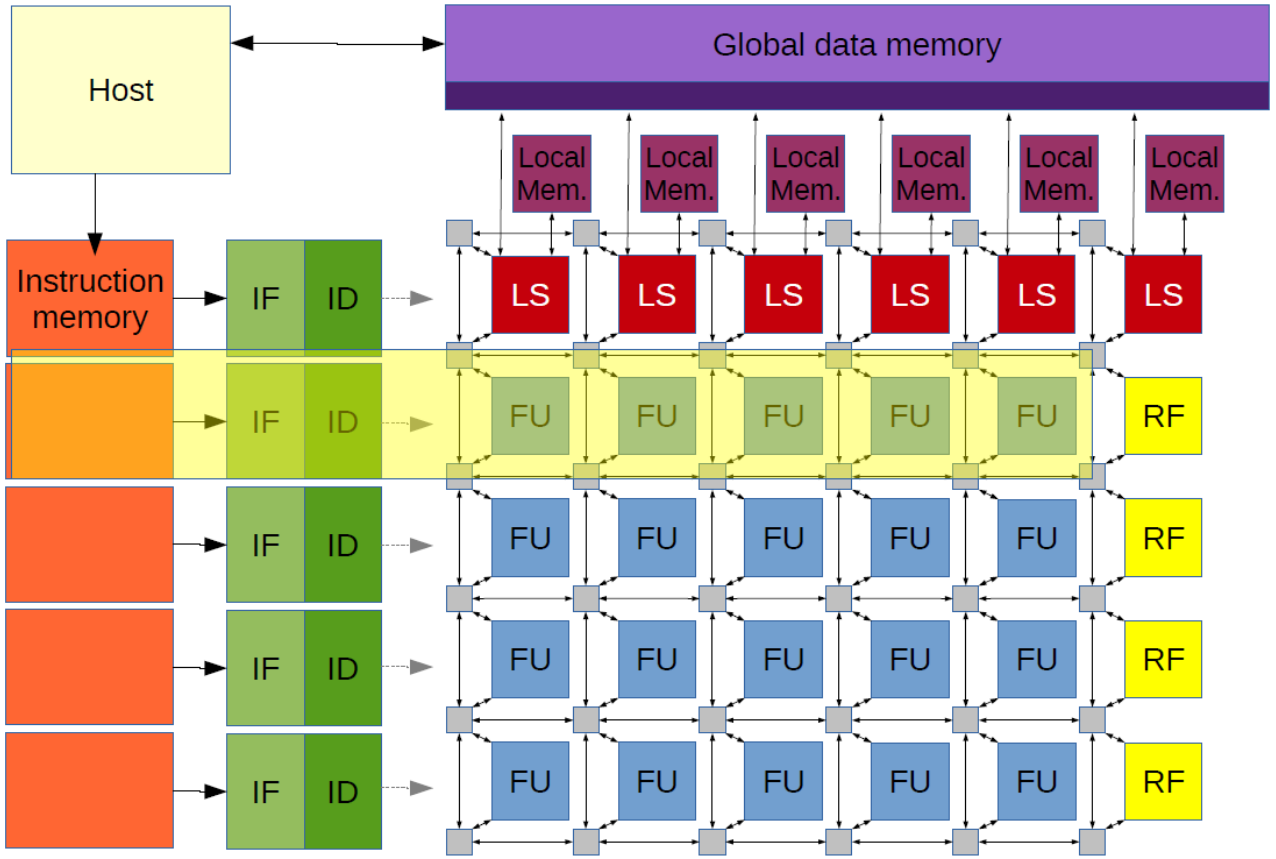
Where can you optimize?

Change the algorithm.

Change the architecture.

Optimized code mapping.

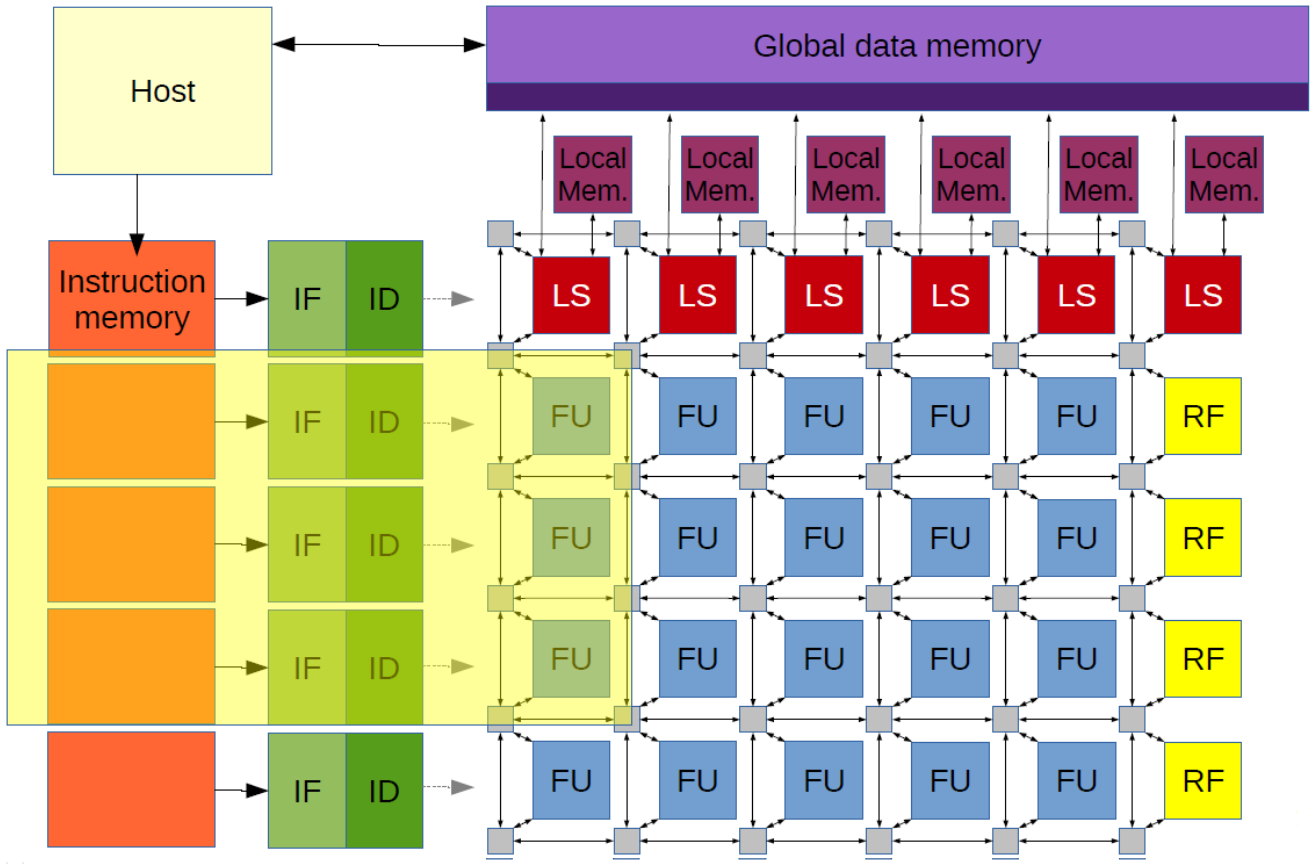
Changing the architecture.



SIMD

Exploit data level parallelism (DLP)

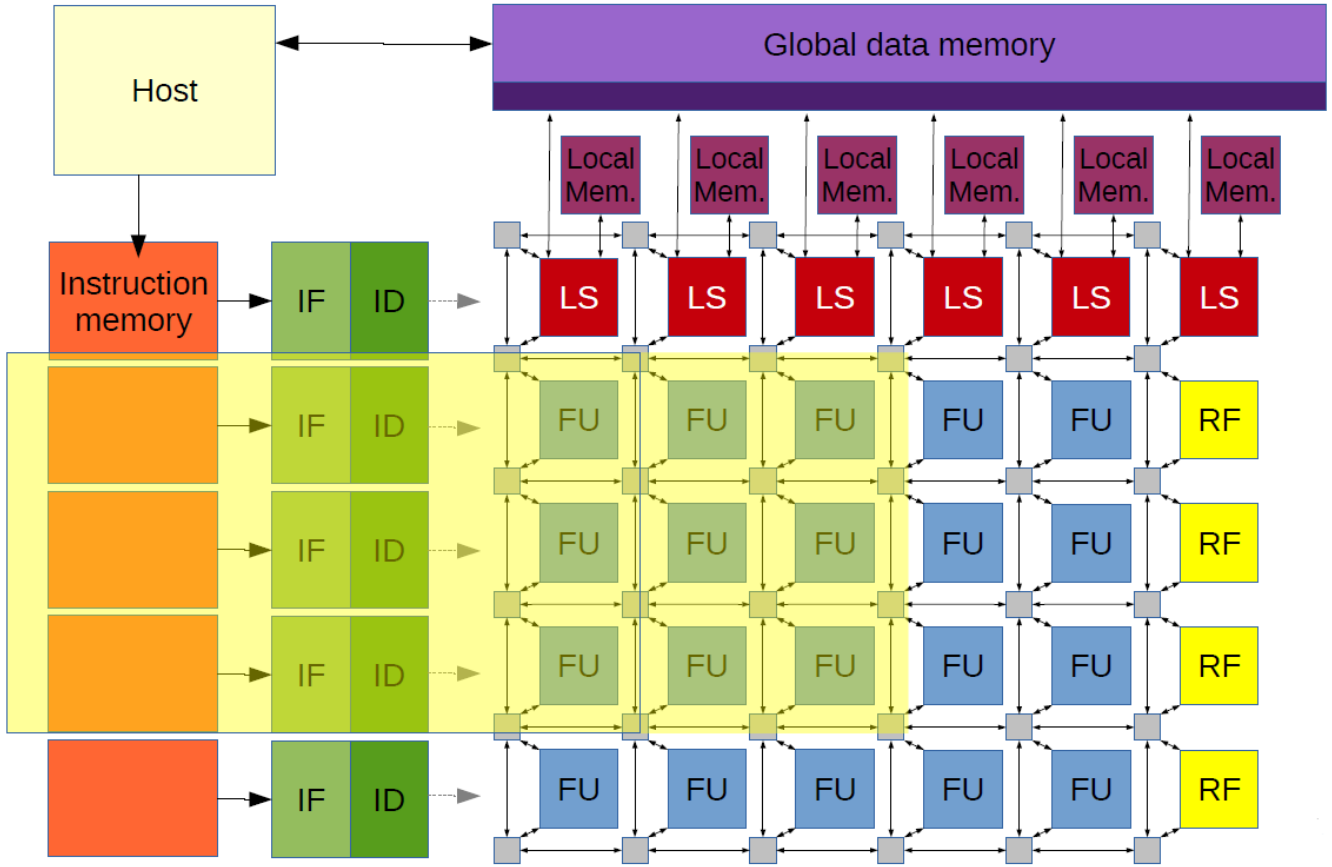
Changing the architecture.



VLIW

Exploit instruction level parallelism (ILP)

Changing the architecture.



ILP + DLP

Changing the architecture.

```
<fu type="ID" name='id_abu'> <!-- IDs -->
    <input index="0" source="abu.1"/>
</fu>
<fu type="ID" name='id_lsu'>
    <input index="0" source="abu.1"/>
</fu>
<fu type="ID" name='id_alu'>
    <input index="0" source="abu.1"/>
</fu>
<fu type="ID" name='id_mul'>
    <input index="0" source="abu.1"/>
</fu>
<fu type="ID" name='id_rf'>
    <input index="0" source="abu.1"/>
</fu>

<fu type="LSU" name='lsu' ID="id_lsu"> <!-- FUs -->
    <input index="0" source="imm.0"/>
    <input index="1" source="rf.1"/>
    <input index="2" source="rf.0"/>
</fu>
<fu type="ALU" name='alu' ID="id_alu" config="1">
    <input index="0" source="rf.0"/>
    <input index="1" source="rf.1"/>
</fu>
<fu type="MUL" name='mul' ID="id_mul" config="1">
    <input index="0" source="rf.0"/>
    <input index="1" source="rf.1"/>
</fu>
<fu type="RF" name='rf' ID="id_rf">
    <input index="0" source="imm.0"/>
    <input index="1" source="lsu.1"/>
    <input index="2" source="alu.1"/>
    <input index="3" source="mul.0"/>
</fu>
<fu type="ABU" name='abu' ID="id_abu" config="1">
    <input index="0" source="alu.1"/>
    <input index="1" source="rf.1"/>
</fu>
<fu type="IU" name='imm'>
    <input index="0" source="abu.1"/>
</fu>
```

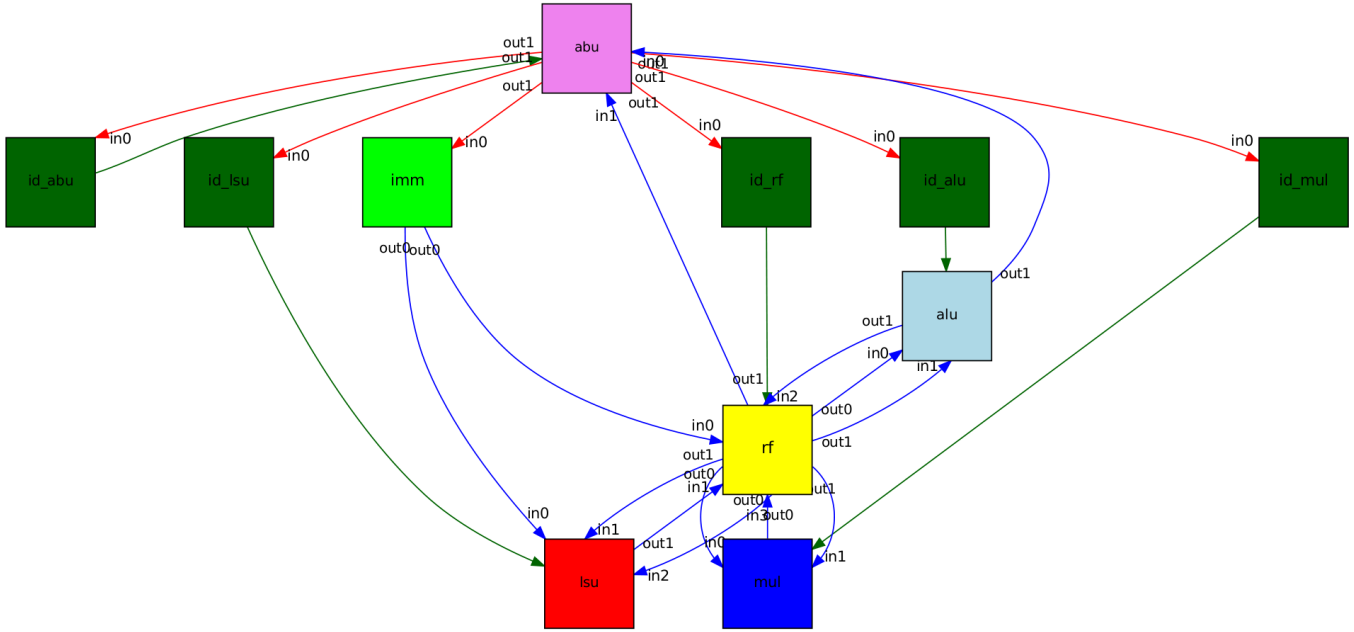
Changing the architecture.

```
<fu type="ID" name='id_abu'> <!-- IDs -->
    <input index="0" source="abu.1"/>
</fu>
<fu type="ID" name='id_lsu'>
    <input index="0" source="abu.1"/>
</fu>
<fu type="ID" name='id_alu'>
    <input index="0" source="abu.1"/>
</fu>
<fu type="ID" name='id_mul'>
    <input index="0" source="abu.1"/>
</fu>
<fu type="ID" name='id_rf'>
    <input index="0" source="abu.1"/>
</fu>

<fu type="LSU" name='lsu' ID="id_lsu"> <!-- FUs -->
    <input index="0" source="imm.0"/>
    <input index="1" source="rf.1"/>
    <input index="2" source="rf.0"/>
</fu>
<fu type="ALU" name='alu' ID="id_alu" config="1">
    <input index="0" source="rf.0"/>
    <input index="1" source="rf.1"/>
</fu>
<fu type="MUL" name='mul' ID="id_mul" config="1">
    <input index="0" source="rf.0"/>
    <input index="1" source="rf.1"/>
</fu>
<fu type="RF" name='rf' ID="id_rf">
    <input index="0" source="imm.0"/>
    <input index="1" source="lsu.1"/>
    <input index="2" source="alu.1"/>
    <input index="3" source="mul.0"/>
</fu>
<fu type="ABU" name='abu' ID="id_abu" config="1">
    <input index="0" source="alu.1"/>
    <input index="1" source="rf.1"/>
</fu>
<fu type="IU" name='imm'>
    <input index="0" source="abu.1"/>
</fu>
```

Graphical view.

$ xdot ~/CGRA/build/convolution/<name>/report/architecture.dot



Changing the architecture.

```
<fu type="ID" name='id_abu'> <!-- IDs -->
    <input index="0" source="abu.1"/>
</fu>
<fu type="ID" name='id_lsu'>
    <input index="0" source="abu.1"/>
</fu>
<fu type="ID" name='id_alu'>
    <input index="0" source="abu.1"/>
</fu>
<fu type="ID" name='id_mul'>
    <input index="0" source="abu.1"/>
</fu>
<fu type="ID" name='id_rf'>
    <input index="0" source="abu.1"/>
</fu>

<fu type="LSU" name='lsu' ID="id_lsu"> <!-- FUs -->
    <input index="0" source="imm.0"/>
    <input index="1" source="rf.1"/>
    <input index="2" source="rf.0"/>
</fu>
<fu type="ALU" name='alu' ID="id_alu" config="1">
    <input index="0" source="rf.0"/>
    <input index="1" source="rf.1"/>
</fu>
<fu type="MUL" name='mul' ID="id_mul" config="1">
    <input index="0" source="rf.0"/>
    <input index="1" source="rf.1"/>
</fu>
<fu type="RF" name='rf' ID="id_rf">
    <input index="0" source="imm.0"/>
    <input index="1" source="lsu.1"/>
    <input index="2" source="alu.1"/>
    <input index="3" source="mul.0"/>
</fu>
<fu type="ABU" name='abu' ID="id_abu" config="1">
    <input index="0" source="alu.1"/>
    <input index="1" source="rf.1"/>
</fu>
<fu type="IU" name='imm'>
    <input index="0" source="abu.1"/>
</fu>
```

Important points.

1. Each ID must have a unique name.
2. All IDs must be connected to the ABU, for the program to advance.
3. FUs are connected to their corresponding IDs.

Optimizing code mapping.

```
| id_abu        || id_rf                  | id_lsu              | imm         | id_alu                | id_mul        |
|               |                          |                     |             |                       |               |
| .text         | .text                    | .text               | .text       | .text                 | .text         |
|               |                          |                     |             |                       |               |
| nop           | nop                      | nop                 | nopi        | nop                   | nop           |
|               |                          |                     | imm 0       |                       |               |
|               | srm r0, in0              |                     | imm 4       |                       |               |
|               | srm r1, in0              |                     | imm 65536   |                       |               |
|               | srm r2, in0              |                     | nopi        |                       |               |
|               | lrm r2                   | sla WORD, in2, in1  | imm 0       |                       |               |
|               | lrm_srm r1, r2, in0      |                     | nopi        | add out1, in0, in1    |               |
|               | srm r0, in2              |                     | nopi        |                       |               |
|               | lrm r2                   | sla WORD, in2, in1  | imm -64572  |                       |               |
|               | lrm_srm r1, r2, in0      |                     | nopi        | add out1, in0, in1    |               |
|               | srm r0, in2              |                     | nopi        |                       |               |
|               | lrm r2                   | sla WORD, in2, in1  | imm 0       |                       |               |
|               | lrm_srm r1, r2, in0      |                     | nopi        | add out1, in0, in1    |               |
|               | srm r0, in2              |                     | nopi        |                       |               |
|               | lrm r2                   | sla WORD, in2, in1  | imm 63818   |                       |               |
|               | lrm_srm r1, r2, in0      |                     | nopi        | add out1, in0, in1    |               |
|               | srm r0, in2              |                     | nopi        |                       |               |
|               | lrm r2                   | sla WORD, in2, in1  | imm 0       |                       |               |
|               | lrm_srm r1, r2, in0      |                     | nopi        | add out1, in0, in1    |               |
|               | srm r0, in2              |                     | nopi        |                       |               |
|               | lrm r2                   | sla WORD, in2, in1  | imm -25323  |                       |               |
|               | lrm_srm r1, r2, in0      |                     | nopi        | add out1, in0, in1    |               |
|               | srm r0, in2              |                     | nopi        |                       |               |
|               | lrm r2                   | sla WORD, in2, in1  | imm 0       |                       |               |
|               | lrm_srm r1, r2, in0      |                     | nopi        | add out1, in0, in1    |               |
                                        ........................................

| bcr in1, in0  | lrm_srm r7, r2, in2      |                     | nopi        |                       |               |
|               |                          |                     | nopi        |                       |               |
|               |                          |                     | nopi        |                       |               |
| jai 0         | nop                      | nop                 | nopi        | nop                   | nop           |
```

Parallel Assembly (PASM)

Optimizing code mapping.

id_abu	id_rf	id_lsu	imm	id_alu	id_mul
.text	.text	.text	.text	.text	.text
nop	nop	nop	nopi	nop	nop
			imm 0		
	srm r0, in0		imm 4		
	srm r1, in0		imm 65536		
	srm r2, in0		nopi		
	lrm r2	sla WORD, in2, in1	imm 0		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm -64572		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm 0		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm 63818		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm 0		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm -25323		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm 0		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
.....					
bcr in1, in0	lrm_srm r7, r2, in2		nopi		
			nopi		
			nopi		
jai 0	nop	nop	nopi	nop	nop

Parallel Assembly (PASM)

Important points.

1. All IDs are specified in different columns.

Optimizing code mapping.

id_abu	id_rf	id_lsu	imm	id_alu	id_mul
.text	.text	.text	.text	.text	.text
nop	nop	nop	nopi	nop	nop
			imm 0		
	srm r0, in0		imm 4		
	srm r1, in0		imm 65536		
	srm r2, in0		nopi		
	lrm r2	sla WORD, in2, in1	imm 0		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm -64572		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm 0		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm 63818		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm 0		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm -25323		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm 0		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	.....				
bcr in1, in0	lrm_srm r7, r2, in2		nopi		
			nopi		
			nopi		
jai 0	nop	nop	nopi	nop	nop

Parallel Assembly (PASM)

Important points.

1. All IDs are specified in different columns
2. Instruction execution starts at count 1 , (not count 0)

Optimizing code mapping.

id_abu	id_rf	id_lsu	imm	id_alu	id_mul
.text	.text	.text	.text	.text	.text
nop	nop	nop	nopi	nop	nop
			imm 0		
	srm r0, in0		imm 4		
	srm r1, in0		imm 65536		
	srm r2, in0		nopi		
	lrm r2	sla WORD, in2, in1	imm 0		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm -64572		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm 0		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm 63818		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm 0		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm -25323		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
	srm r0, in2		nopi		
	lrm r2	sla WORD, in2, in1	imm 0		
	lrm_srm r1, r2, in0		nopi	add out1, in0, in1	
.....					
bcr in1, in0	lrm_srm r7, r2, in2		nopi		
			nopi		
			nopi		
jai 0	nop	nop	nopi	nop	nop

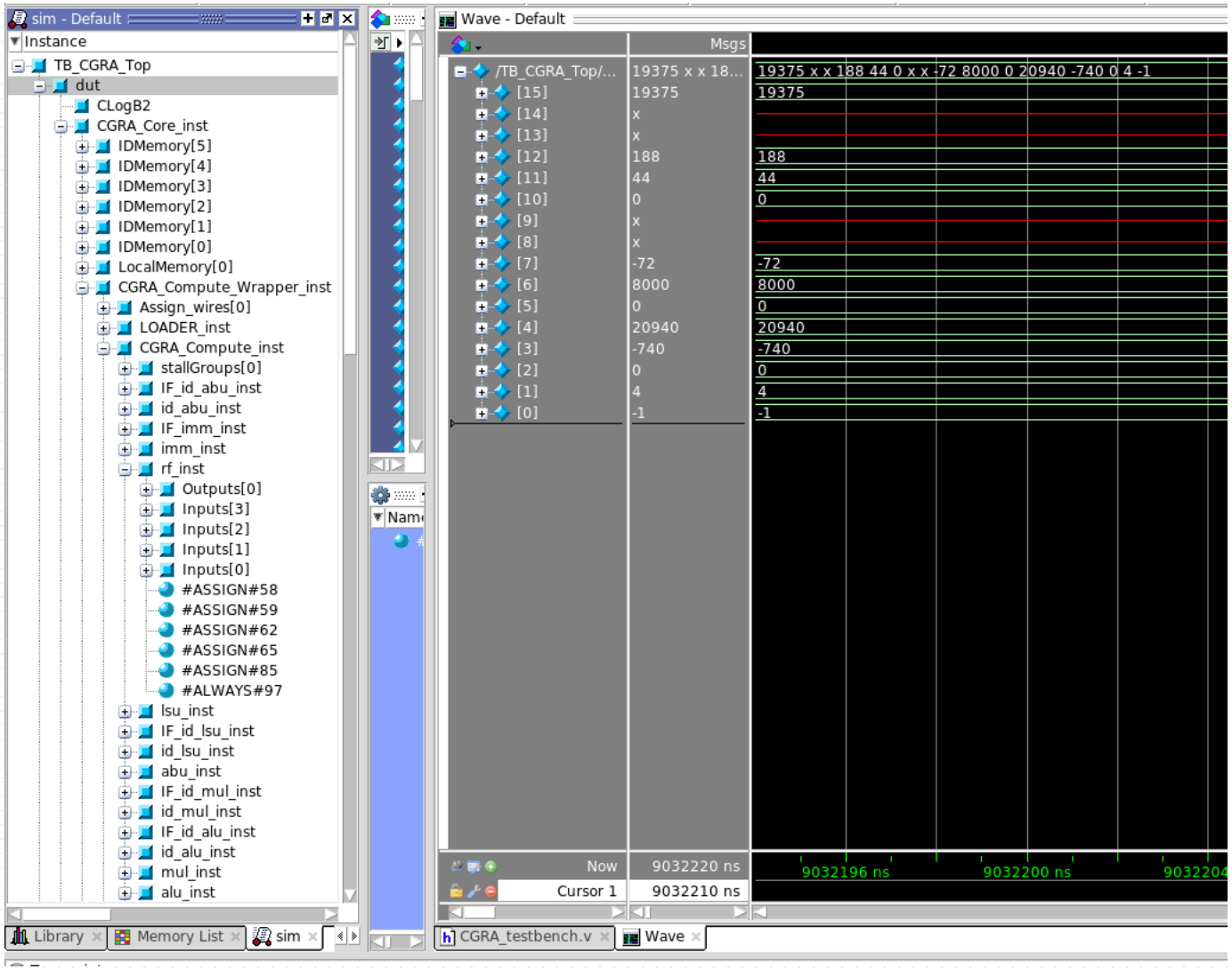
Parallel Assembly (PASM)

Important points.

1. All IDs are specified in different columns
2. Instruction execution starts at count 1 , (not count 0)
3. “jai 0” ends program execution.

Detailed ISA can be found on https://oncourse.tue.nl

Debugging



Use Modelsim.

```
$ cd ~/CGRA/benchmarks/convolution/<name>
$ make sim
```

To add register file :

TB_CGRA_Top

↓

dut

↓

CGRA_Core_inst

↓

CGRA_Compute_Wrapper_inst

↓

CGRA_Compute_inst

↓

rf_inst

Makefile

make sim

Runs modelsim with GUI.

make run

Runs modelsim in the command line.

make compare

Runs simulation and compares the output with actual one (given in compare directory).

make performance

Runs simulation and gives quick performance estimates (less accurate).

make report

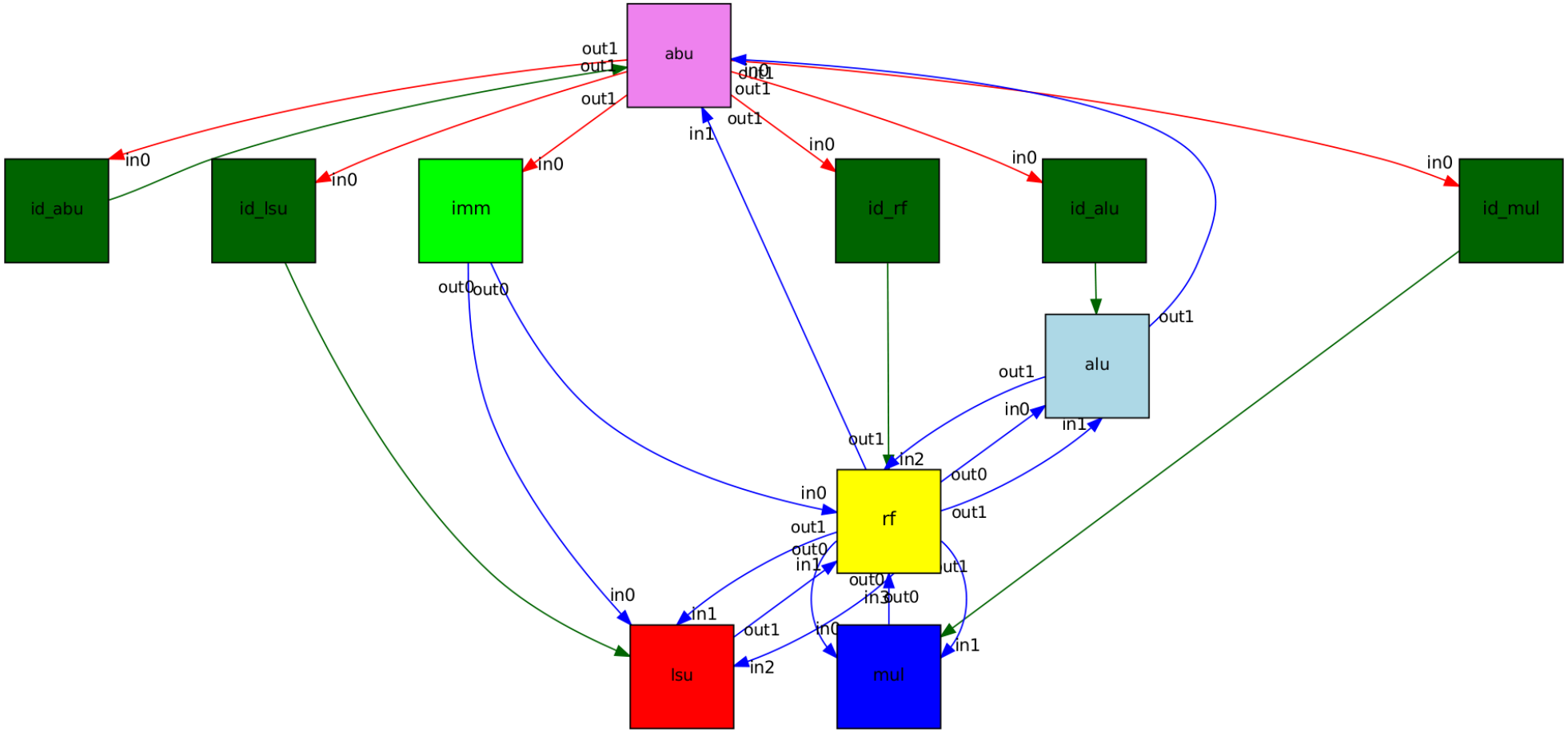
Runs simulation and generates a report in build/<name>/report (slower but accurate).

make clean

Cleans the build directory for the benchmark.

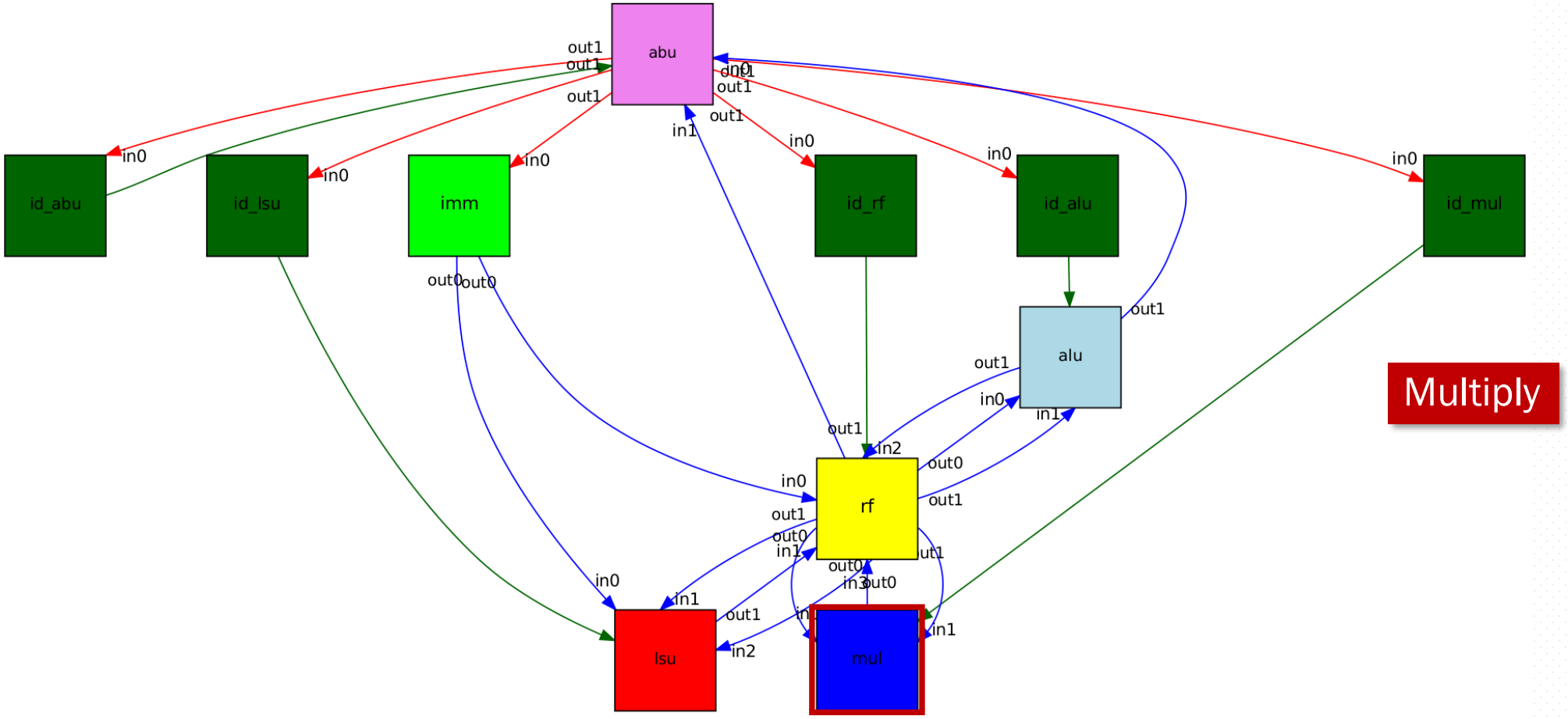
Optimization Hints

Bypassing



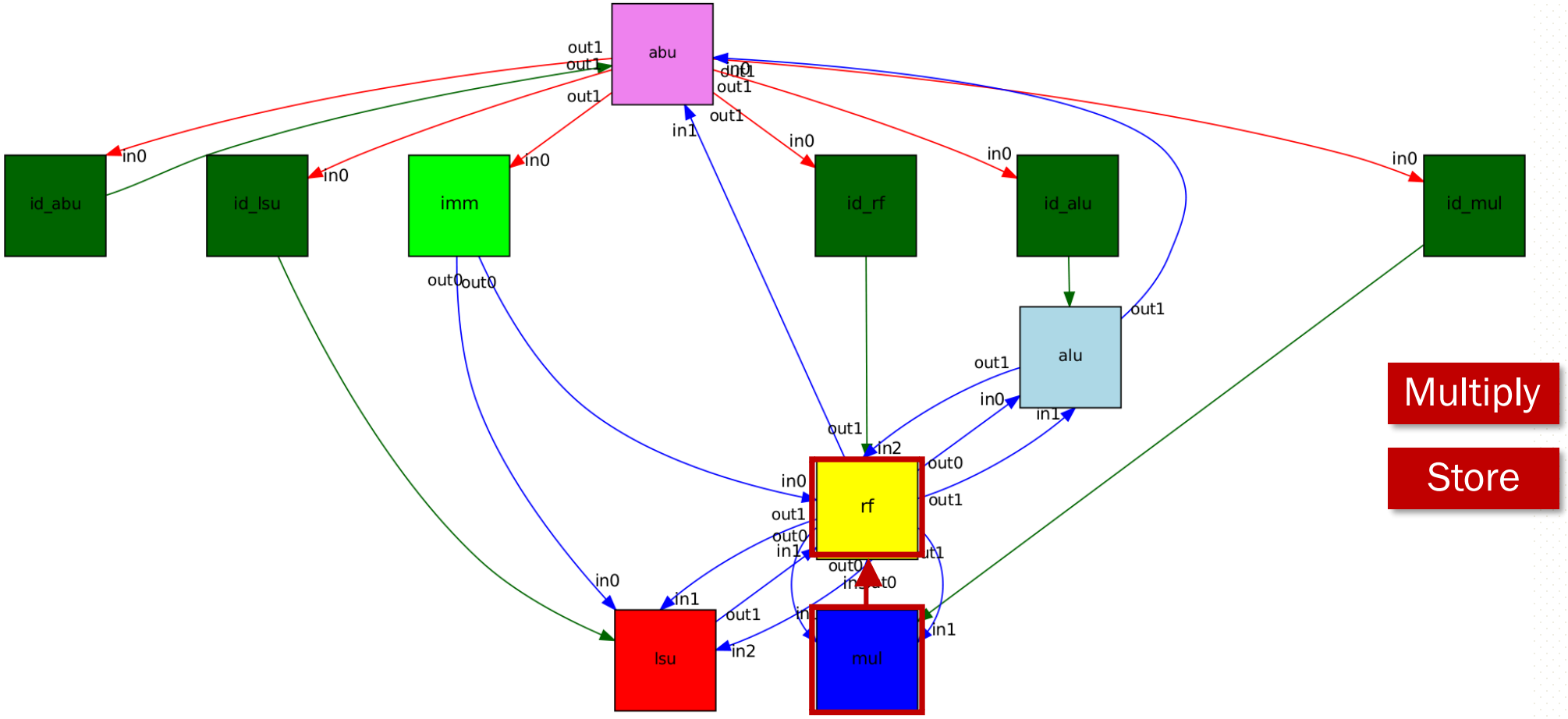
Optimization Hints

Bypassing



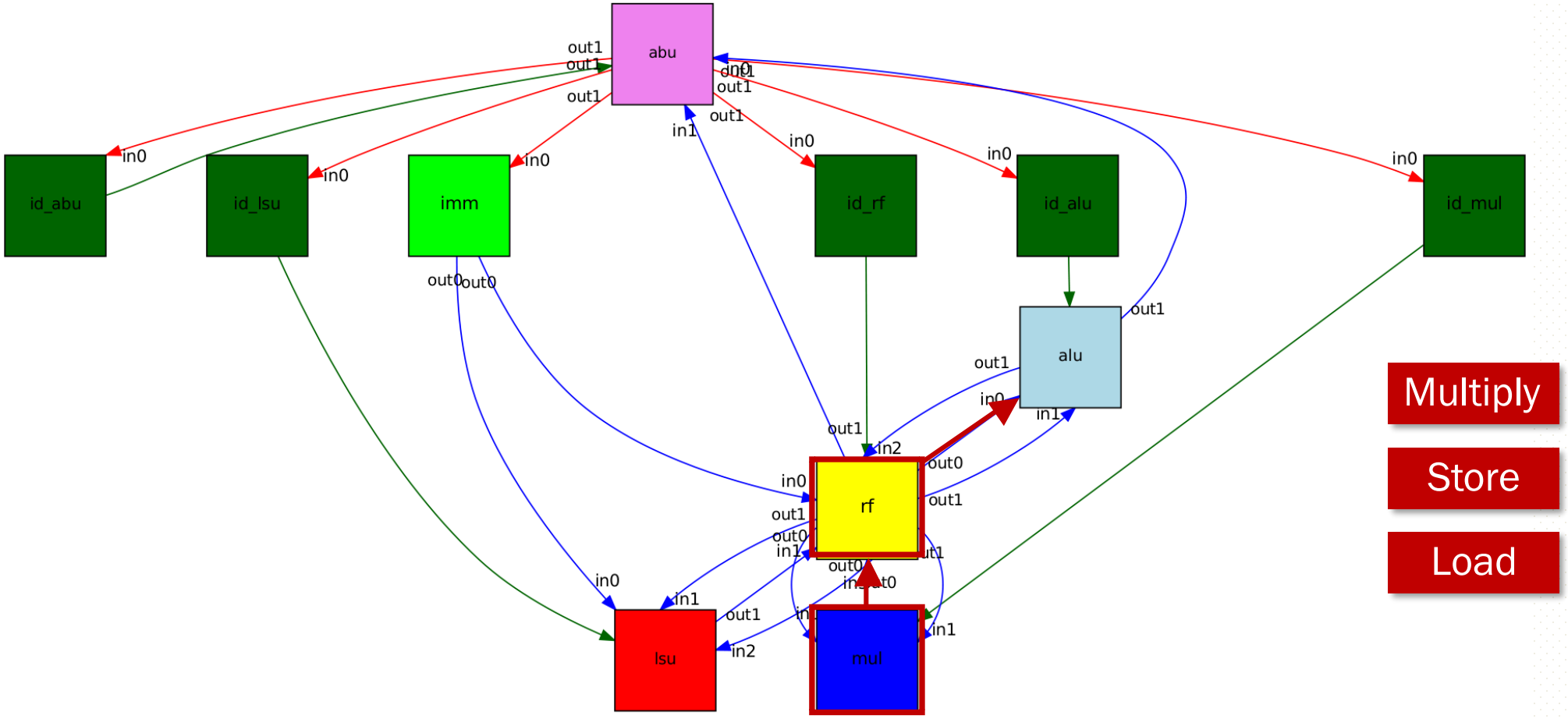
Optimization Hints

Bypassing



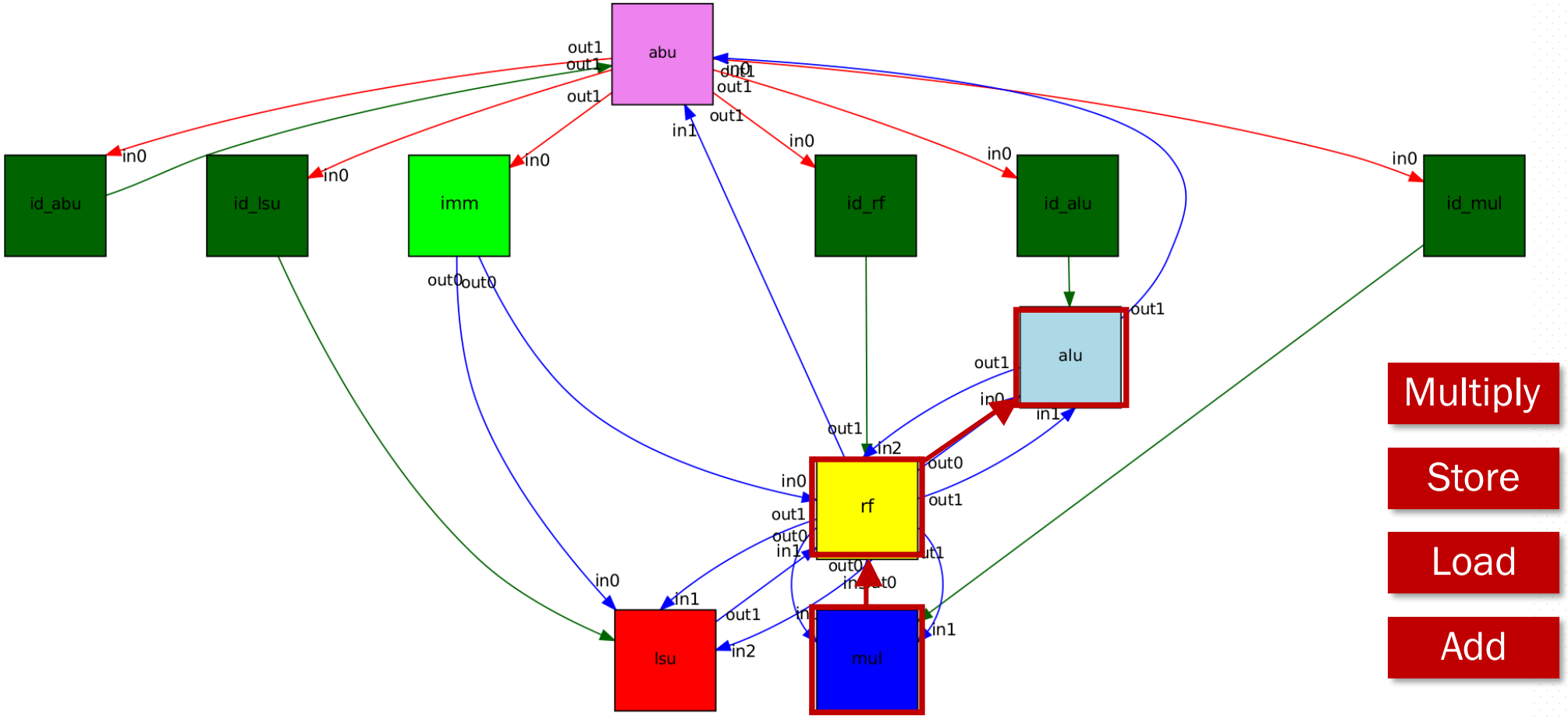
Optimization Hints

Bypassing



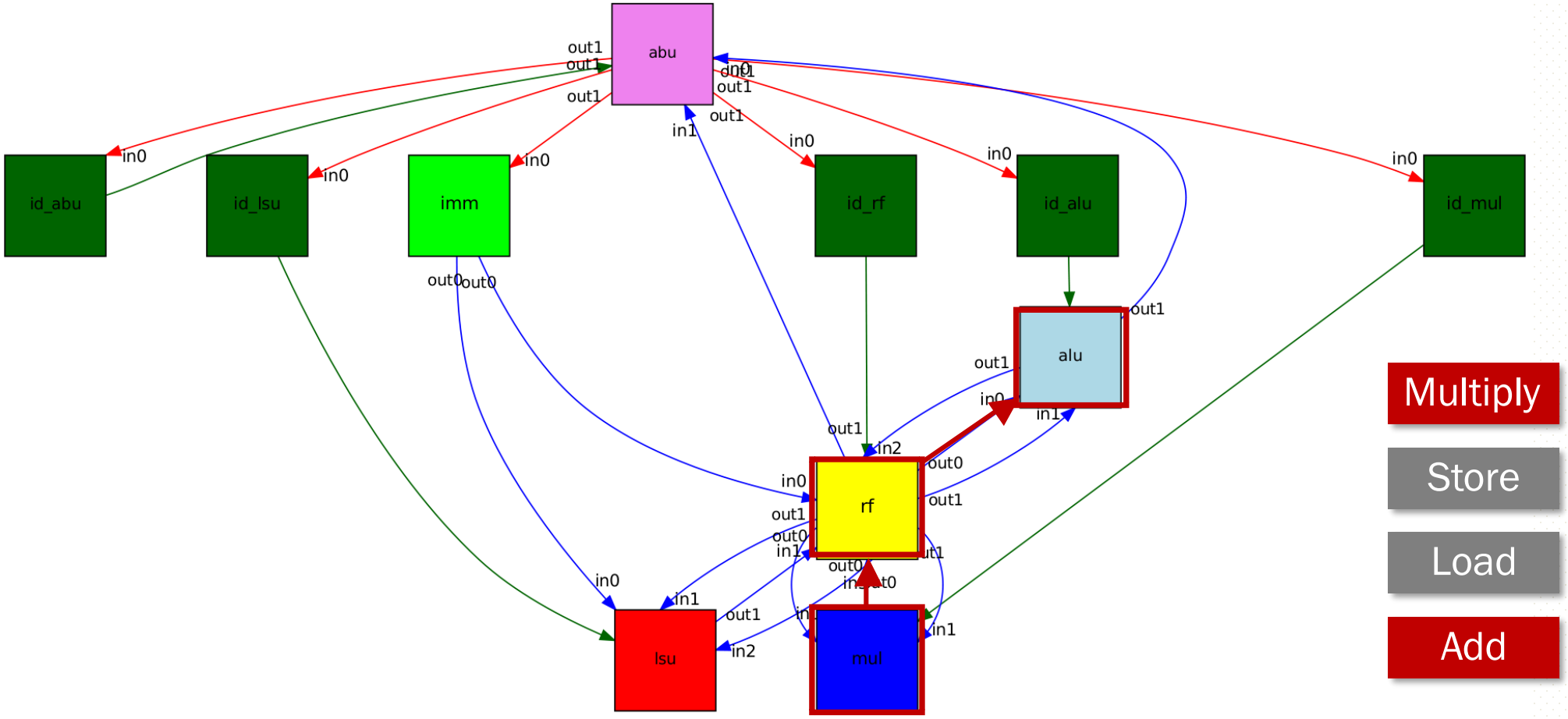
Optimization Hints

Bypassing



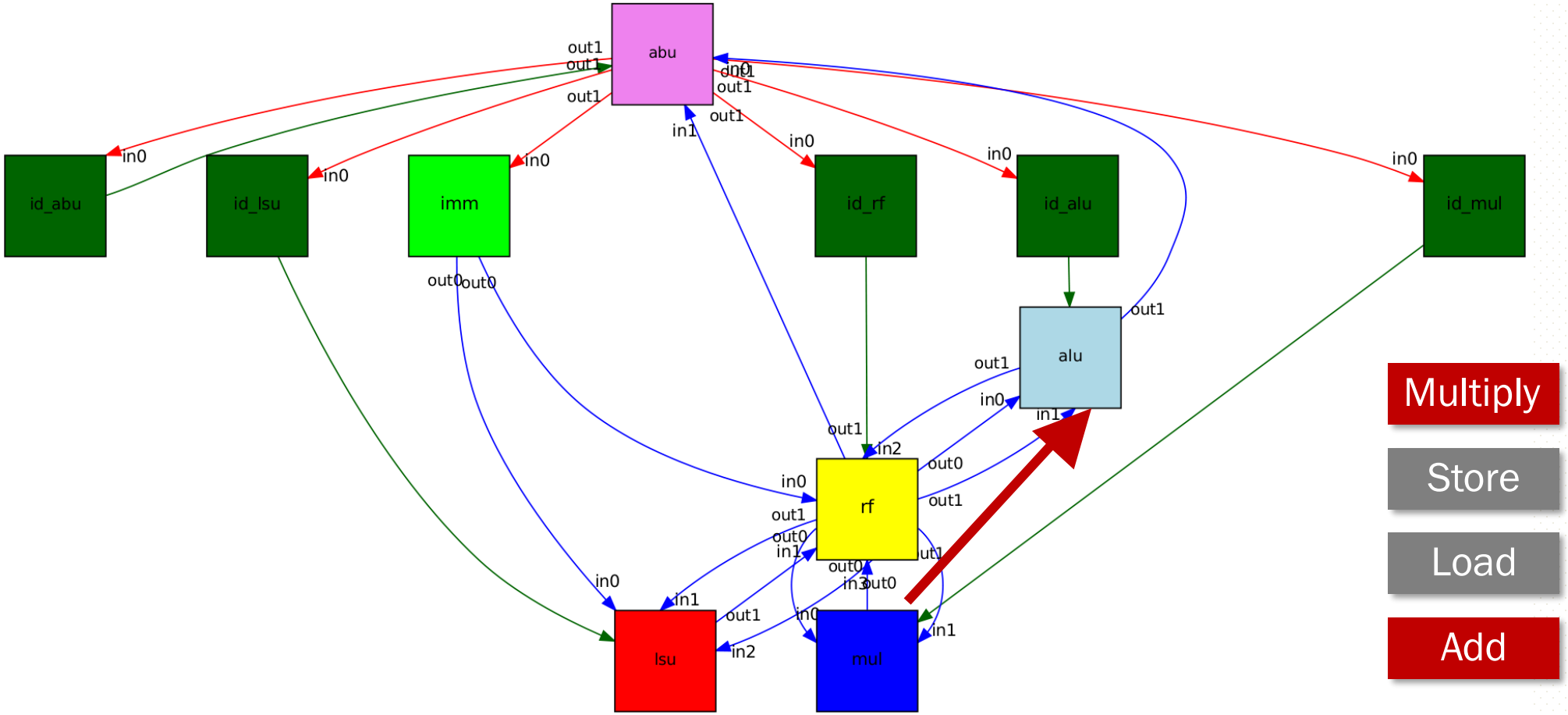
Optimization Hints

Bypassing



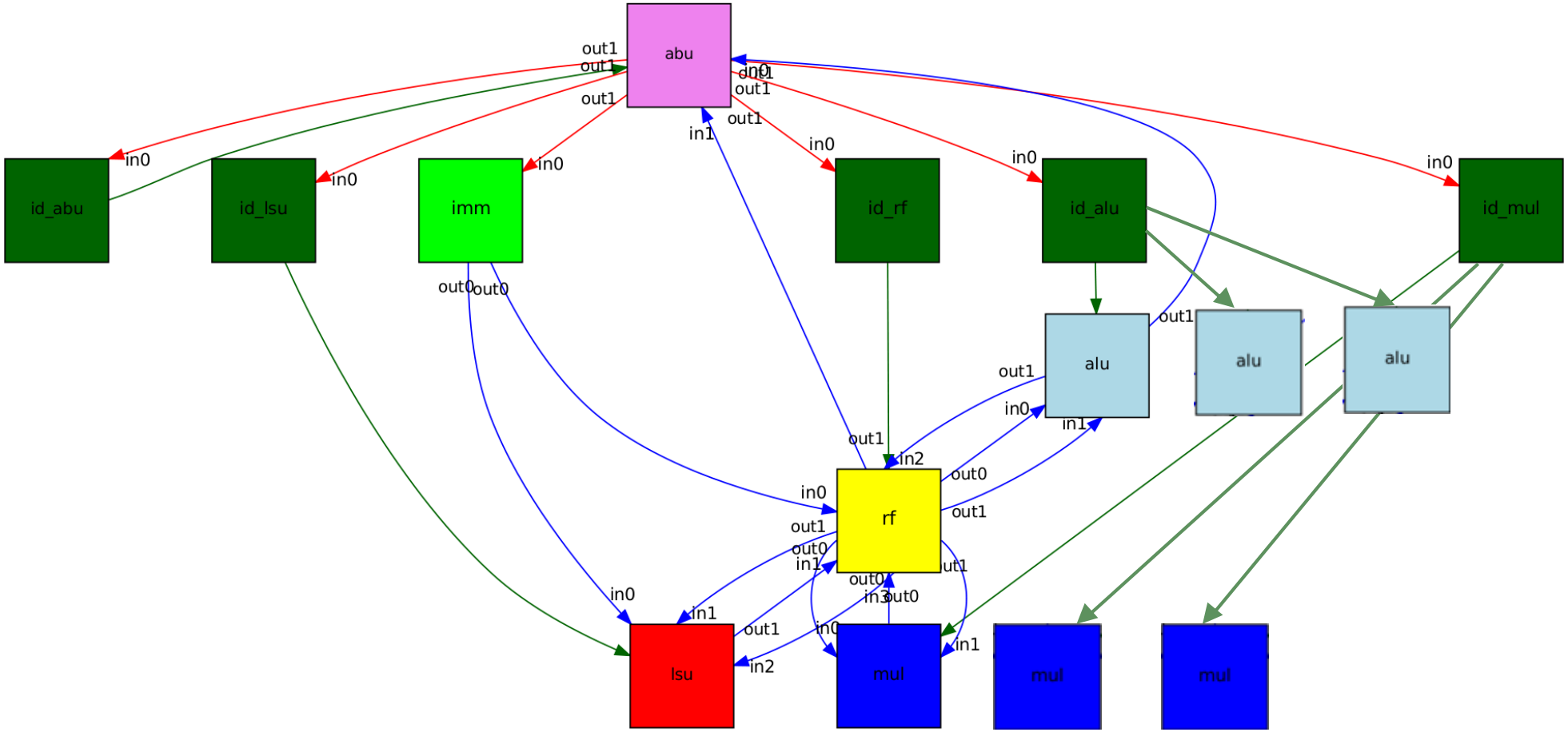
Optimization Hints

Bypassing



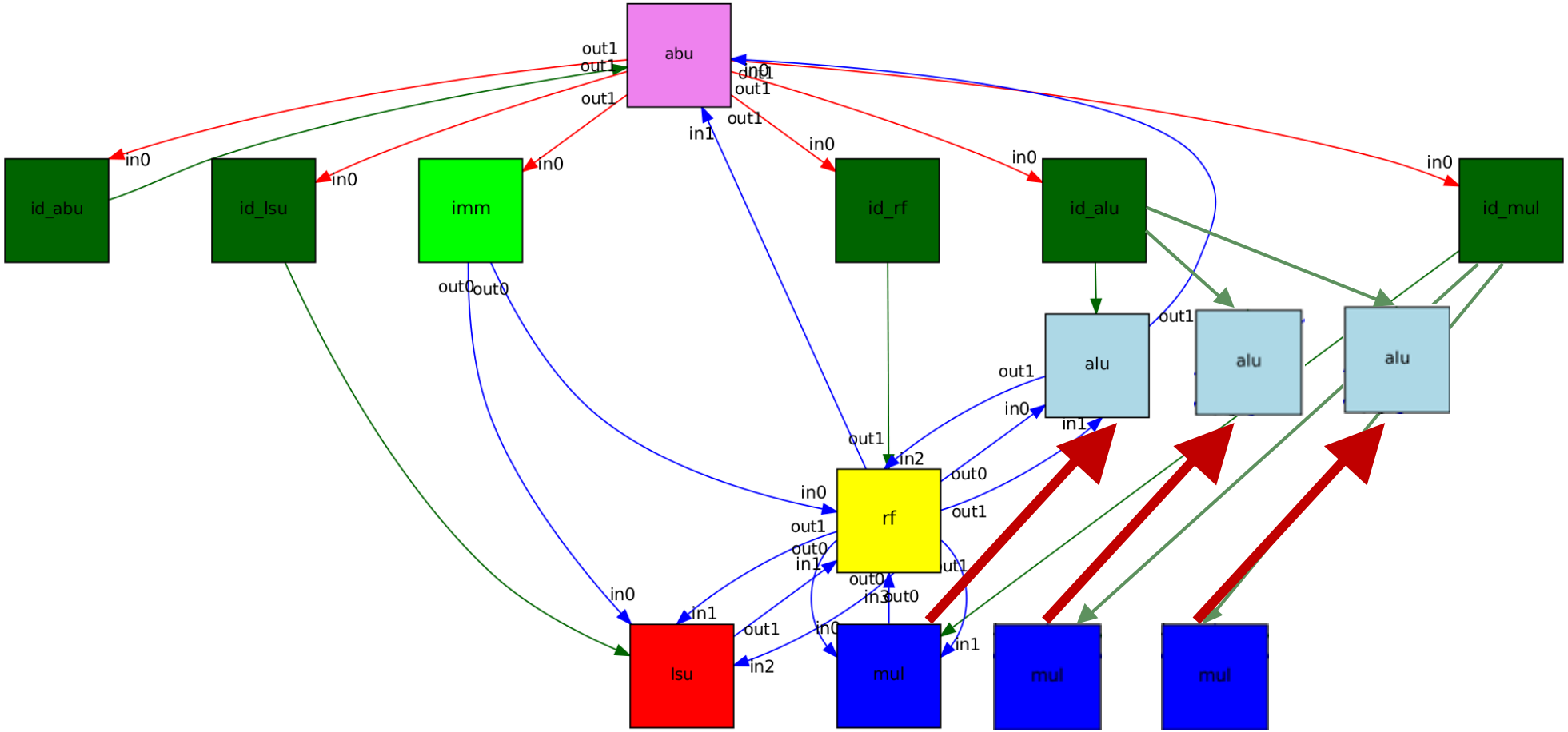
Optimization Hints

SIMD



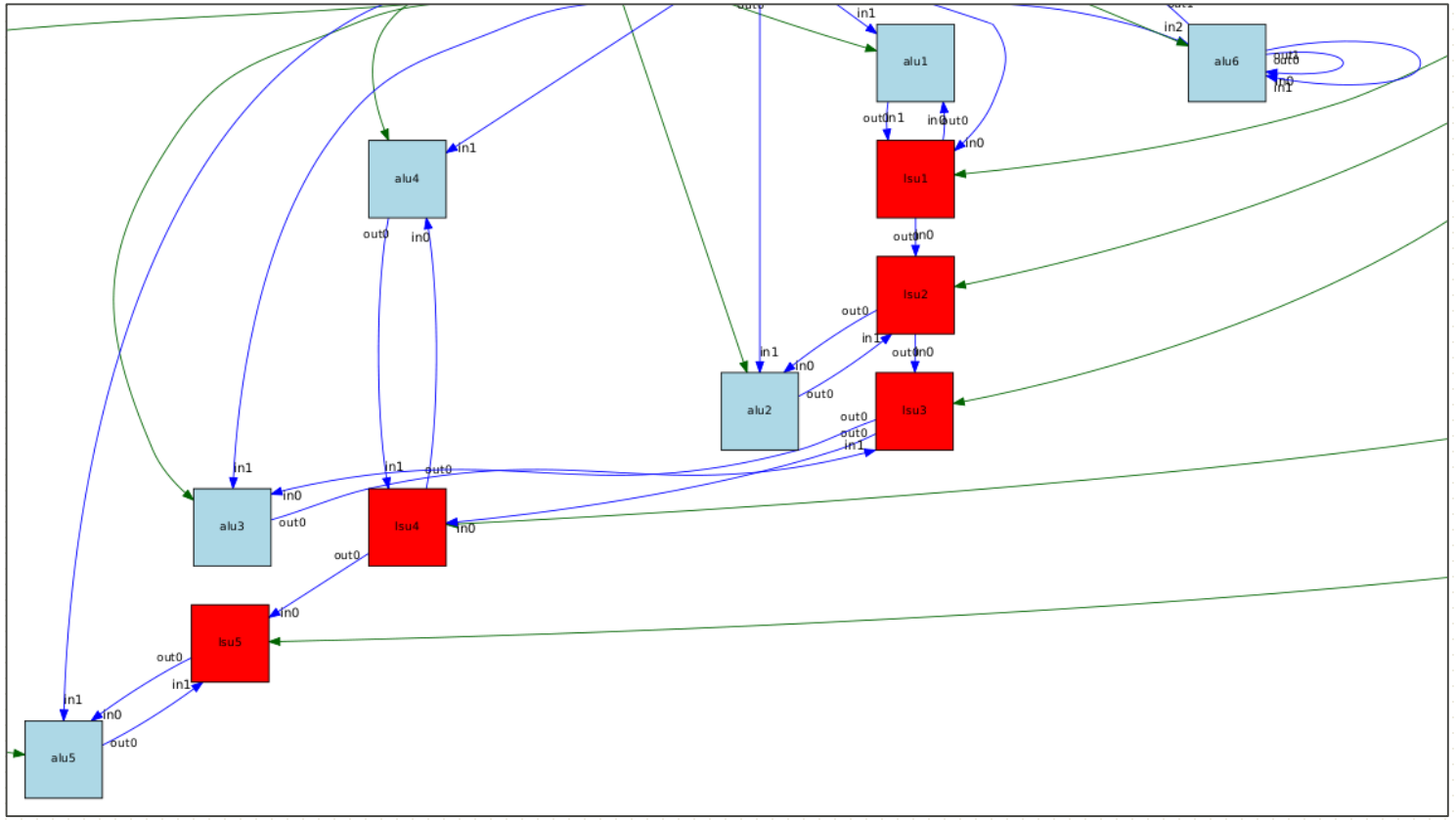
Optimization Hints

Bypassing + SIMD



Optimization Hints

Single Cycle Loop

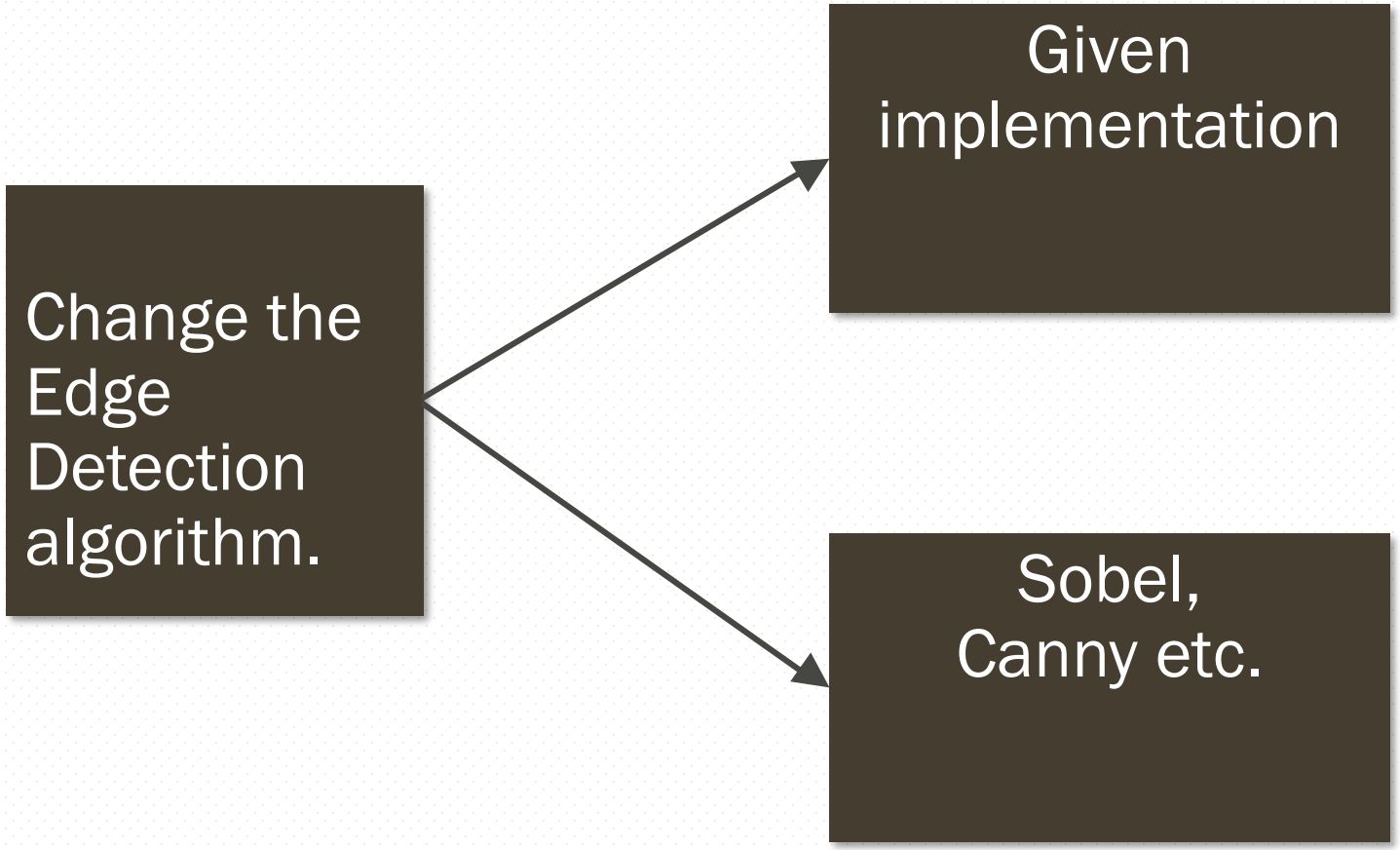



```
| id_abu                                  | id_lsu                          | imm                                 | id_alu1
|                                         |                                 |                                     |
                                                                                                        .
| nop                                     | lgi BYTE, out0                  | imm 128                             | nop
| nop                                     | lgi BYTE, out0                  | imm 128                             | ltu out0, in0, in1
| bcai 20, in0                            | lgi_sgi BYTE, out0, in1         | imm 128                             | ltu out0, in0, in1
| nop ; delay slot 1                      | lgi_sgi BYTE, out0, in1         | imm 128                             | ltu out0, in0, in1
| nop ; delay slot 2                      | lgi_sgi BYTE, out0, in1         | imm 128                             | ltu out0, in0, in1
| nop                                     | sgi BYTE, in1                   | nopi                                | ltu out0, in0, in1
| nop                                     | sgi BYTE, in1                   | nopi                                | nop
| jai 0 ; terminate execution             | nop                             | nopi                                | nop
|                                         |                                 |                                     |
```

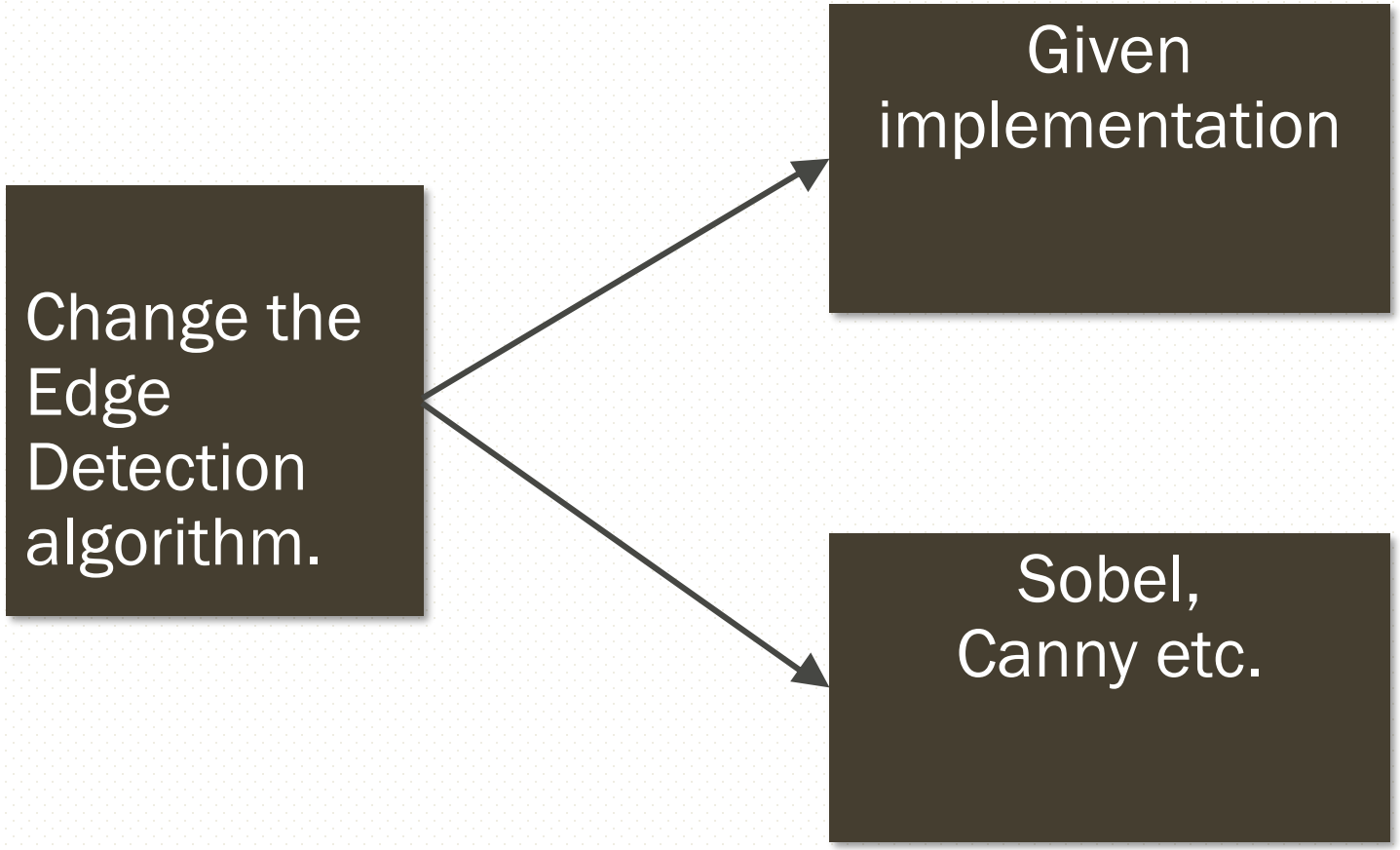
Optimization Hints

- Data reuse
- Automatic address generation
- Adding ABU (for accumulation)

Changing the algorithm



Changing the algorithm



In case, you use a different algorithm, provide us with the C implementation as well.

Server List

co2.ics.ele.tue.nl

co3.ics.ele.tue.nl

co4.ics.ele.tue.nl

co9.ics.ele.tue.nl

co10.ics.ele.tue.nl

co13.ics.ele.tue.nl

co17.ics.ele.tue.nl

Resources

- Assignment details and guideline

https://oncourse.tue.nl