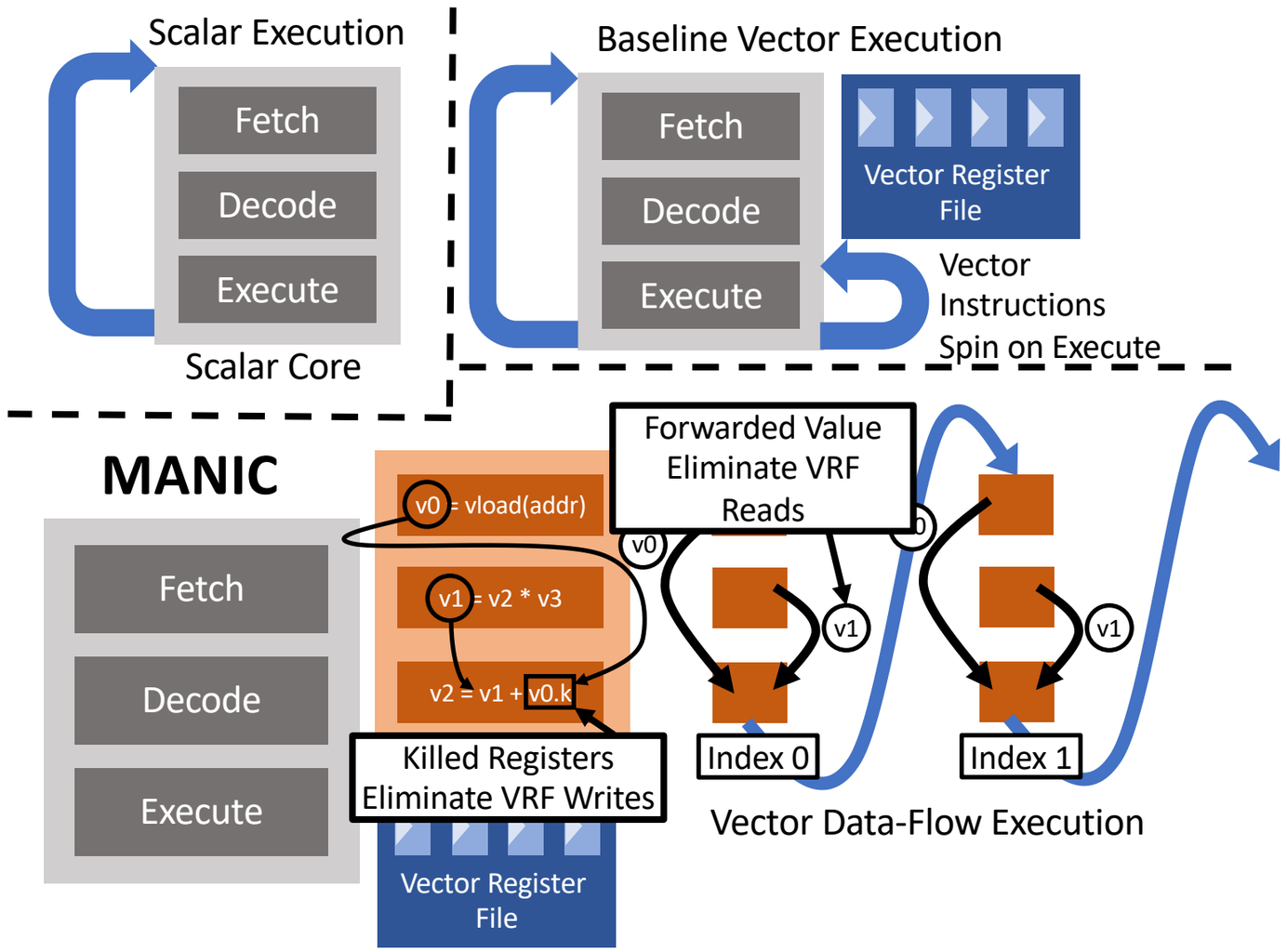
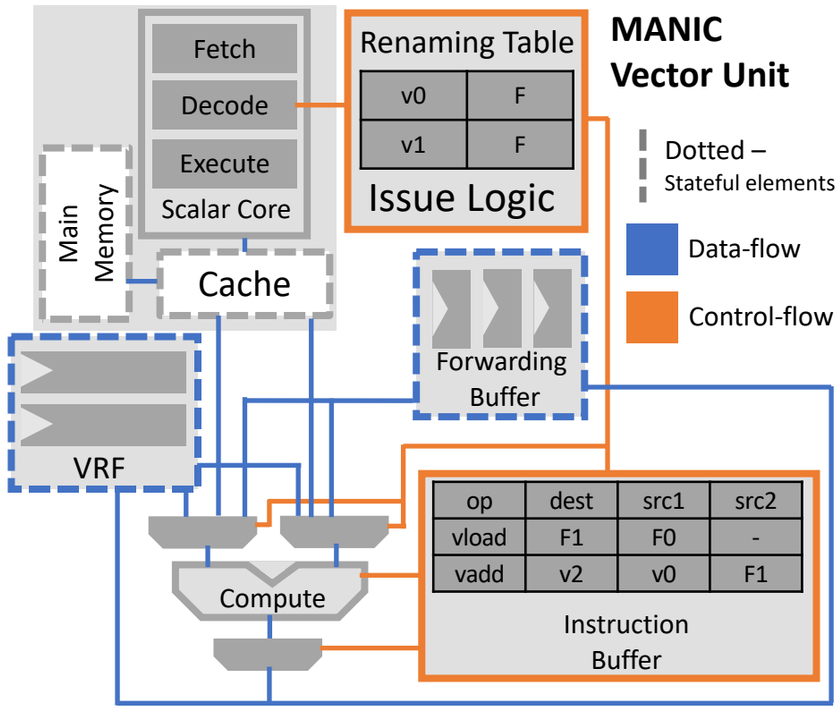
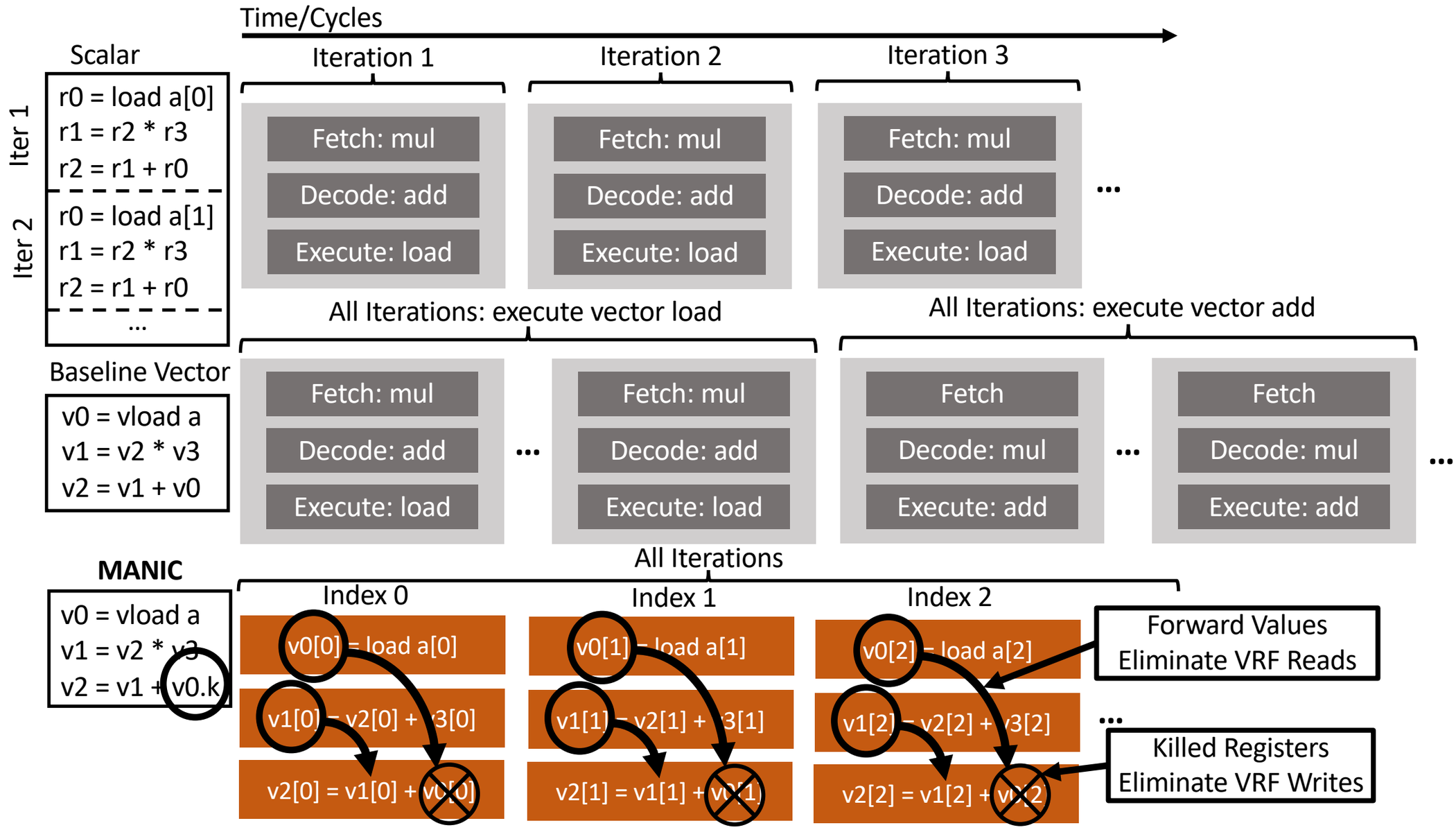
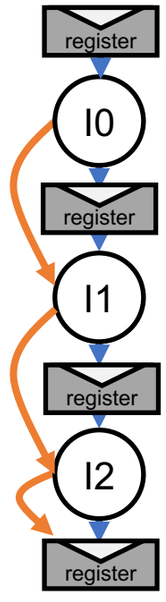
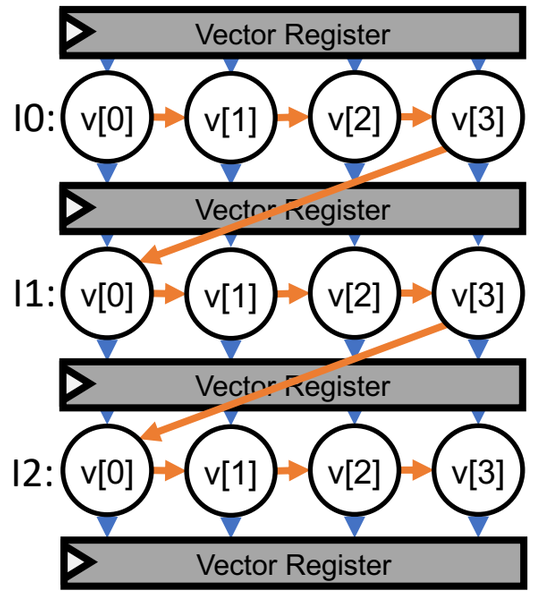
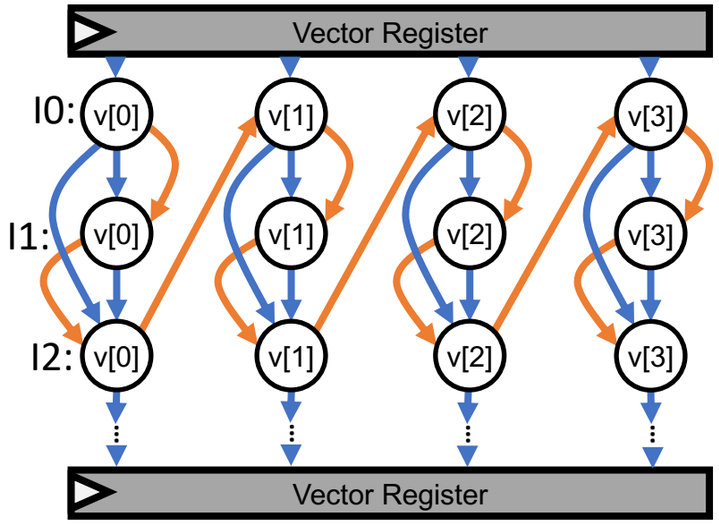

Scalar Execution
Fetch
Decode
Execute
Scalar Core
Baseline Vector Execution
Fetch
Decode
Execute
Vector Register File
Vector Instructions Spin on Execute
MANIC
Fetch
Decode
Execute
v0 = vload(addr)
v1 = v2 * v3
v2 = v1 + v0.k
Killed Registers Eliminate VRF Writes
Vector Register File
Forwarded Value Eliminate VRF Reads
v0
v1
Index 0
Index 1
Vector Data-Flow Execution

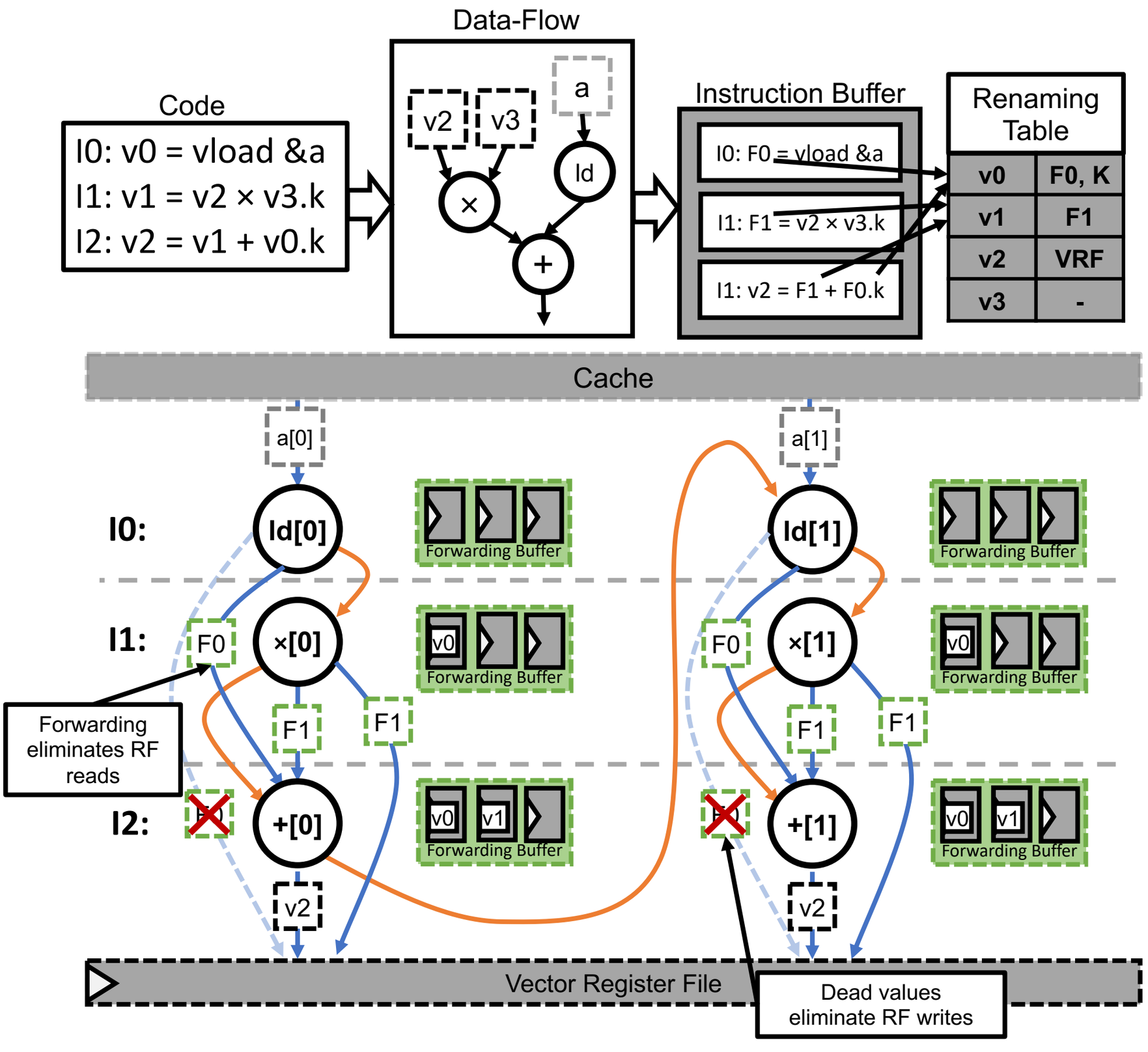

Fetch
Decode
Execute
Scalar Core
Main Memory
Cache
Renaming Table
v0
F
v1
F
Issue Logic
MANIC
Vector Unit
Dotted –
Stateful elements
Data-flow
Control-flow
Forwarding
Buffer
VRF
op
dest
src1
src2
vload
F1
F0
-
vadd
v2
v0
F1
Compute
Instruction
Buffer

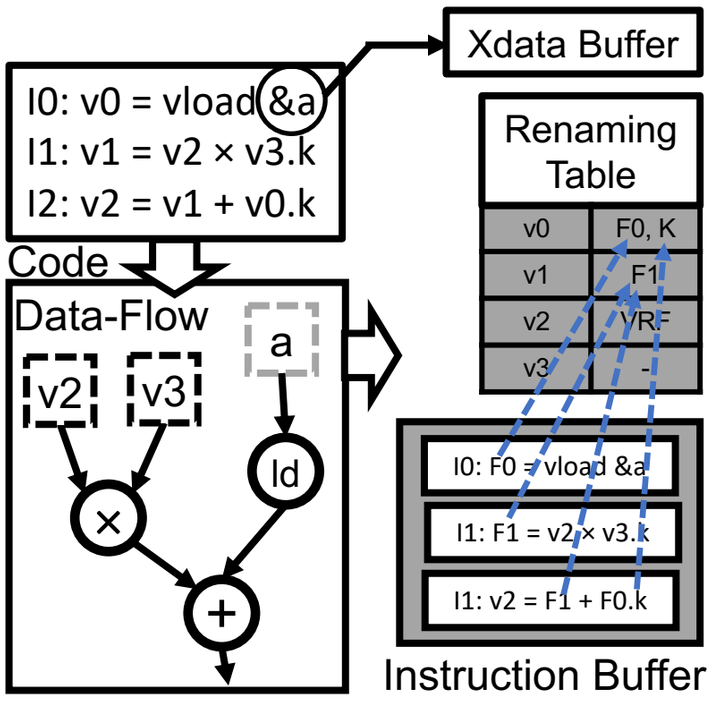

Time/Cycles
Scalar
Iteration 1
Iteration 2
Iteration 3
Iter 1
r0 = load a[0]
r1 = r2 * r3
r2 = r1 + r0
Iter 2
r0 = load a[1]
r1 = r2 * r3
r2 = r1 + r0
...
Fetch: mul
Decode: add
Execute: load
...
All Iterations: execute vector load
All Iterations: execute vector add
Baseline Vector
v0 = vload a
v1 = v2 * v3
v2 = v1 + v0
Fetch: mul
Decode: add
Execute: load
Fetch
Decode: mul
Execute: add
MANIC
v0 = vload a
v1 = v2 * v3
v2 = v1 + v0.k
All Iterations
Index 0
Index 1
Index 2
v0[0] = load a[0]
v0[1] = load a[1]
v0[2] = load a[2]
v1[0] = v2[0] + v3[0]
v1[1] = v2[1] + v3[1]
v1[2] = v2[2] + v3[2]
v2[0] = v1[0] + v0[0]
v2[1] = v1[1] + v0[1]
v2[2] = v1[2] + v0[2]
Forward Values
Eliminate VRF Reads
Killed Registers
Eliminate VRF Writes

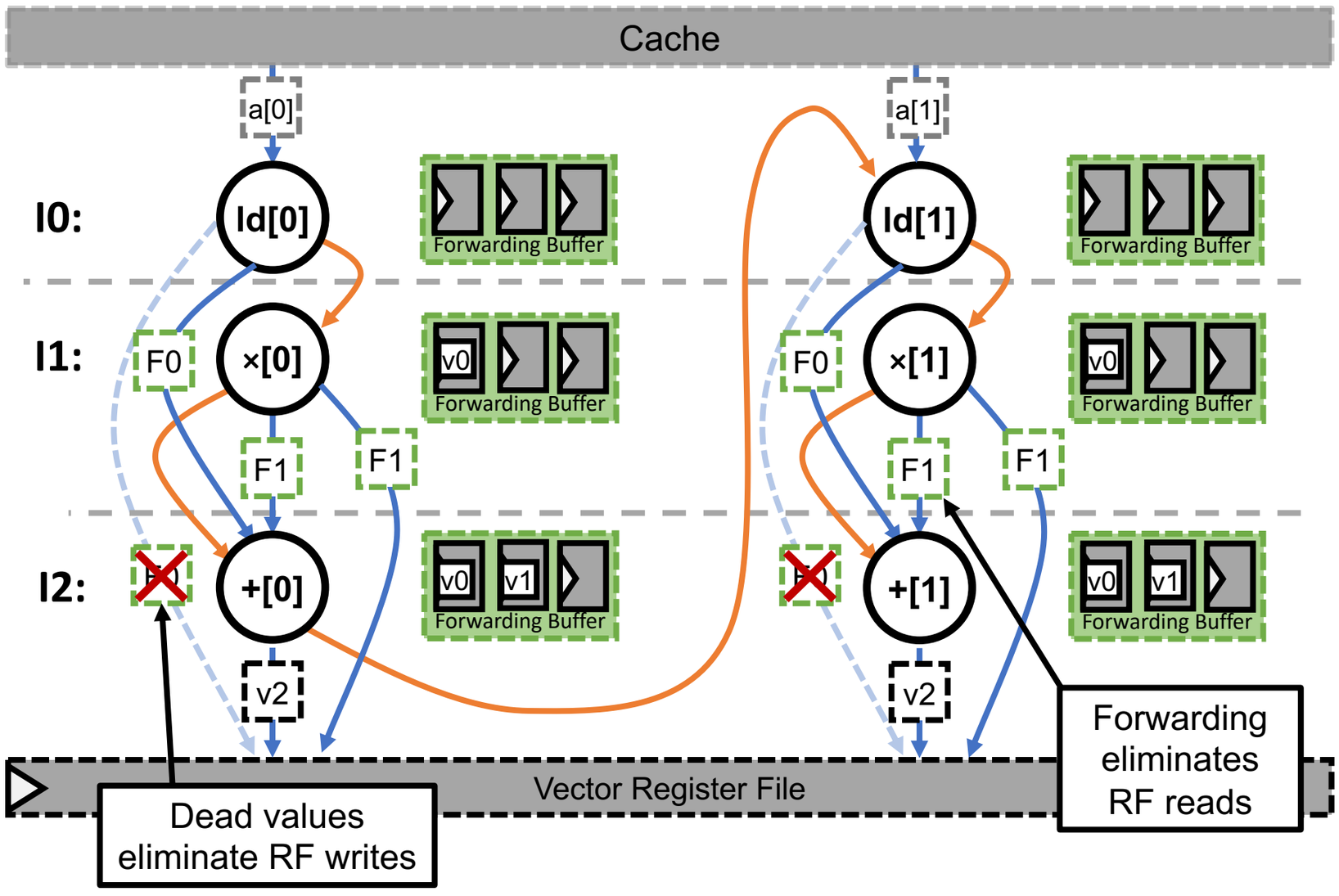

register
I0
register
I1
register
I2
register

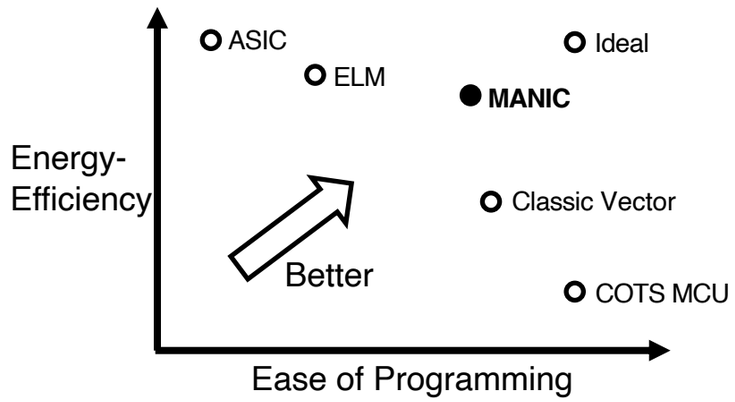

Vector Register
I0:
v[0]
v[1]
v[2]
v[3]
Vector Register
I1:
v[0]
v[1]
v[2]
v[3]
Vector Register
I2:
v[0]
v[1]
v[2]
v[3]
Vector Register

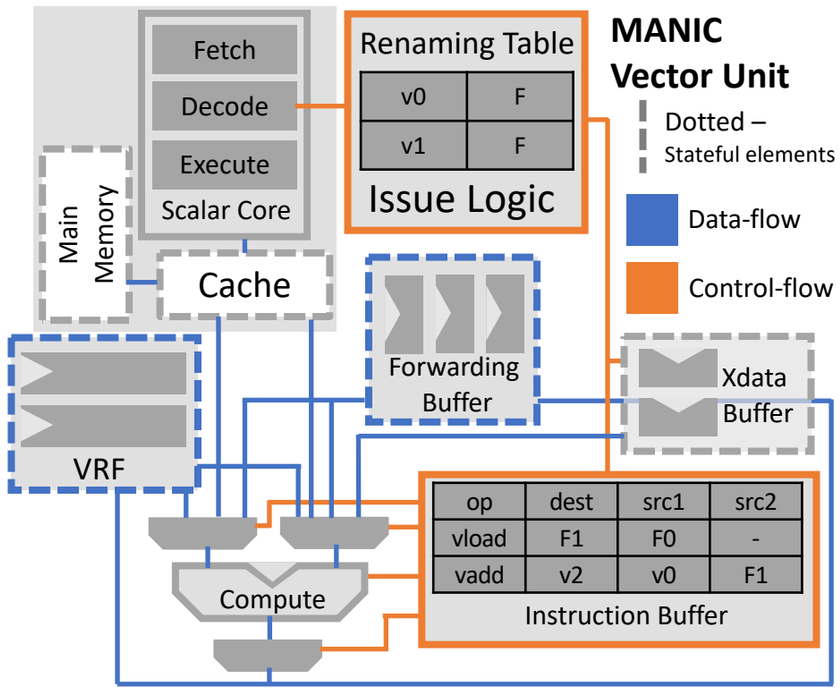

Vector Register
I0:
I1:
I2:
v[0]
v[1]
v[2]
v[3]
Vector Register

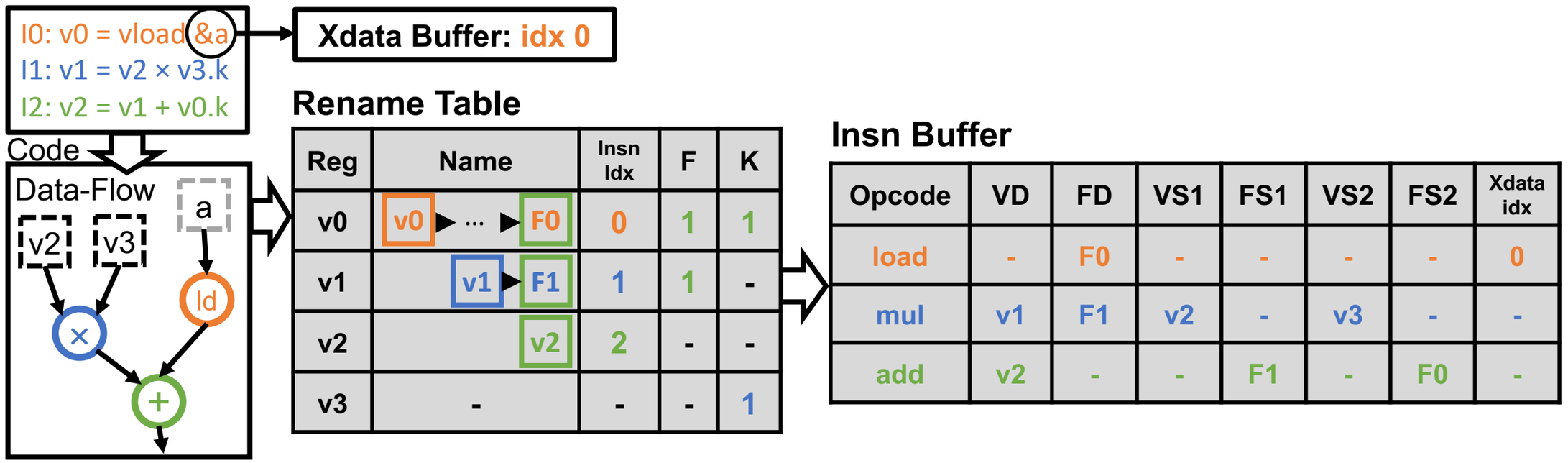

Code
I0: v0 = vload &a
I1: v1 = v2 × v3.k
I2: v2 = v1 + v0.k
Data-Flow
v2
v3
a
ld
×
+
Instruction Buffer
I0: F0 = vload &a
I1: F1 = v2 × v3.k
I1: v2 = F1 + F0.k
Renaming Table
v0
F0, K
v1
F1
v2
VRF
v3
-
Cache
a[0]
a[1]
I0:
ld[0]
ld[1]
Forwarding Buffer
I1:
F0
×[0]
×[1]
v0
F1
I2:
+[0]
+[1]
v1
v2
Forwarding eliminates RF reads
Vector Register File
Dead values eliminate RF writes

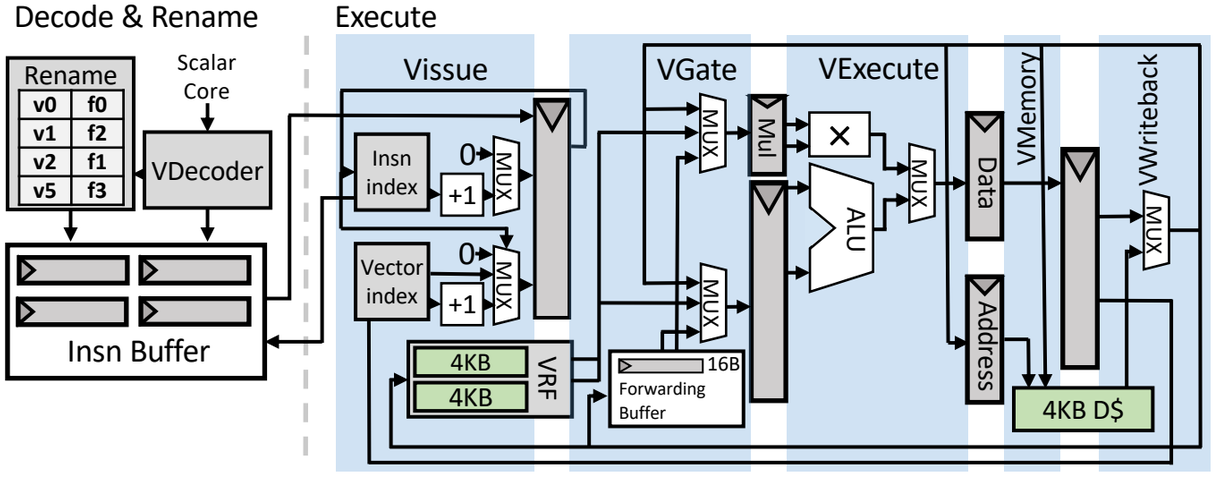

Xdata Buffer
I0: v0 = vload &a
I1: v1 = v2 × v3.k
I2: v2 = v1 + v0.k
Code
Data-Flow
a
v2
v3
ld
×
+
Renaming Table
v0
F0, K
v1
F1
v2
VRF
v3
-
I0: F0 = vload &a
I1: F1 = v2 × v3.k
I1: v2 = F1 + F0.k
Instruction Buffer

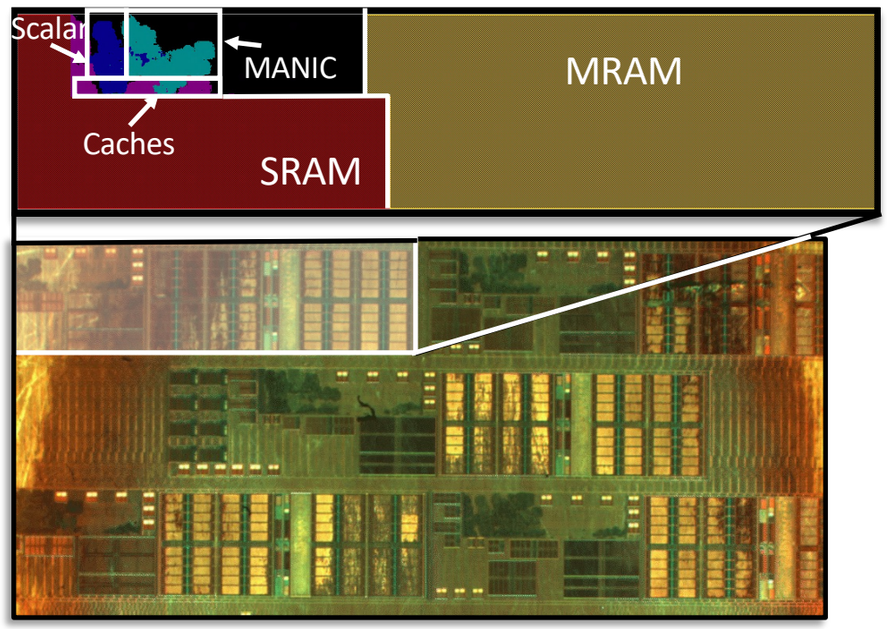

Cache
a[0]
a[1]
I0:
Id[0]
Id[1]
Forwarding Buffer
I1:
F0
×[0]
×[1]
v0
F1
I2:
+[0]
+[1]
v1
v2
Vector Register File
Dead values
eliminate RF writes
Forwarding
eliminates
RF reads

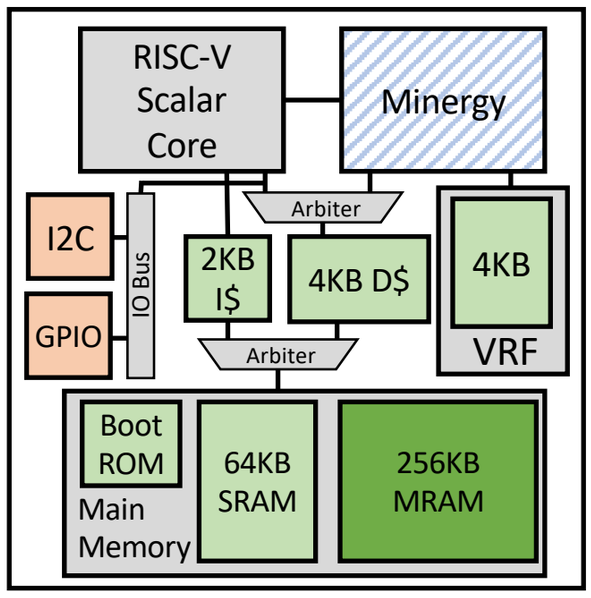

ASIC
Ideal
ELM
MANIC
Energy-
Efficiency
Classic Vector
Better
COTS MCU
Ease of Programming

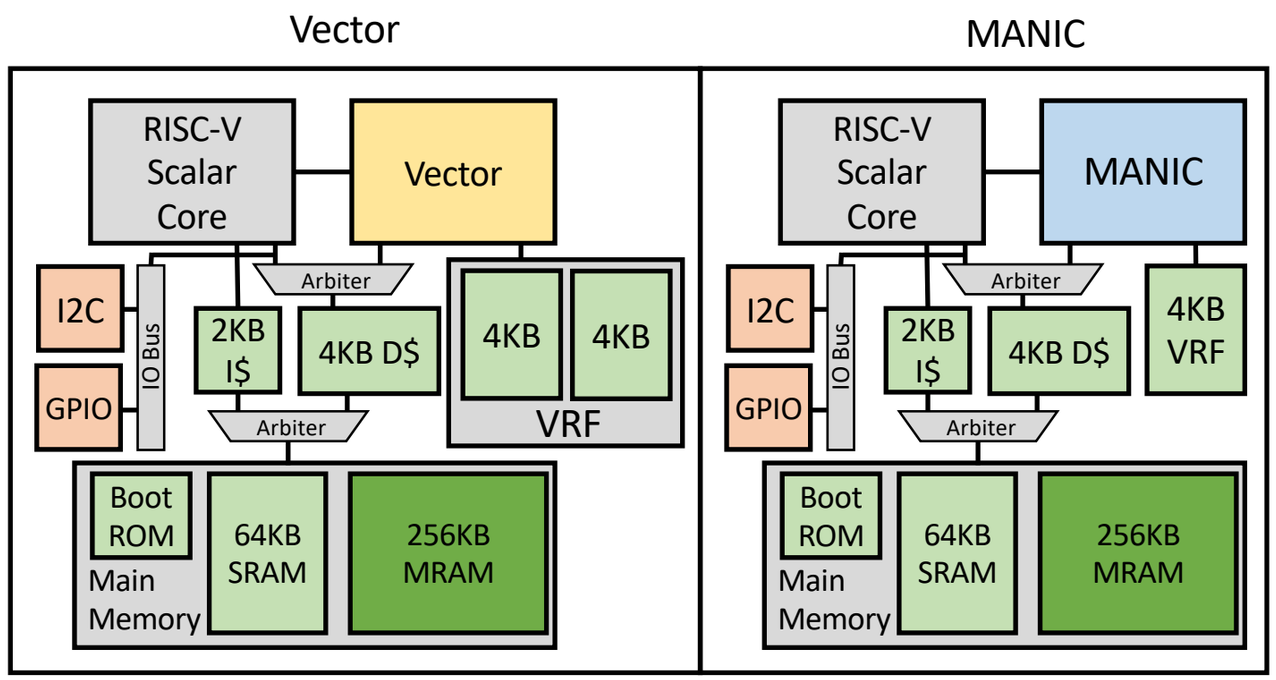

Fetch
Decode
Execute
Scalar Core
Main Memory
Cache
Renaming Table
v0
F
v1
F
Issue Logic
MANIC
Vector Unit
Dotted –
Stateful elements
Data-flow
Control-flow
Forwarding
Buffer
Xdata
Buffer
VRF
op
dest
src1
src2
vload
F1
F0
-
vadd
v2
v0
F1
Instruction Buffer
Compute

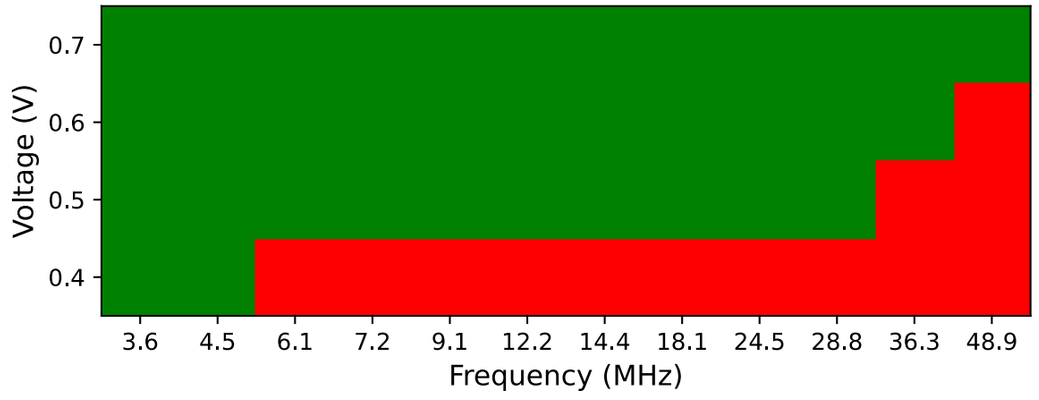

I0: v0 = vload &a
I1: v1 = v2 × v3.k
I2: v2 = v1 + v0.k
Code
Data-Flow
a
v2
v3
ld
×
+
Xdata Buffer: idx 0
Rename Table
Reg
Name
Insn Idx
F
K
v0
v0 ▶ ··· ▶ F0
0
1
1
v1
v1 ▶ F1
1
1
-
v2
v2
2
-
-
v3
-
-
-
1
Insn Buffer
Opcode
VD
FD
VS1
FS1
VS2
FS2
Xdata idx
load
-
F0
-
-
-
-
0
mul
v1
F1
v2
-
v3
-
-
add
v2
-
-
F1
-
F0
-

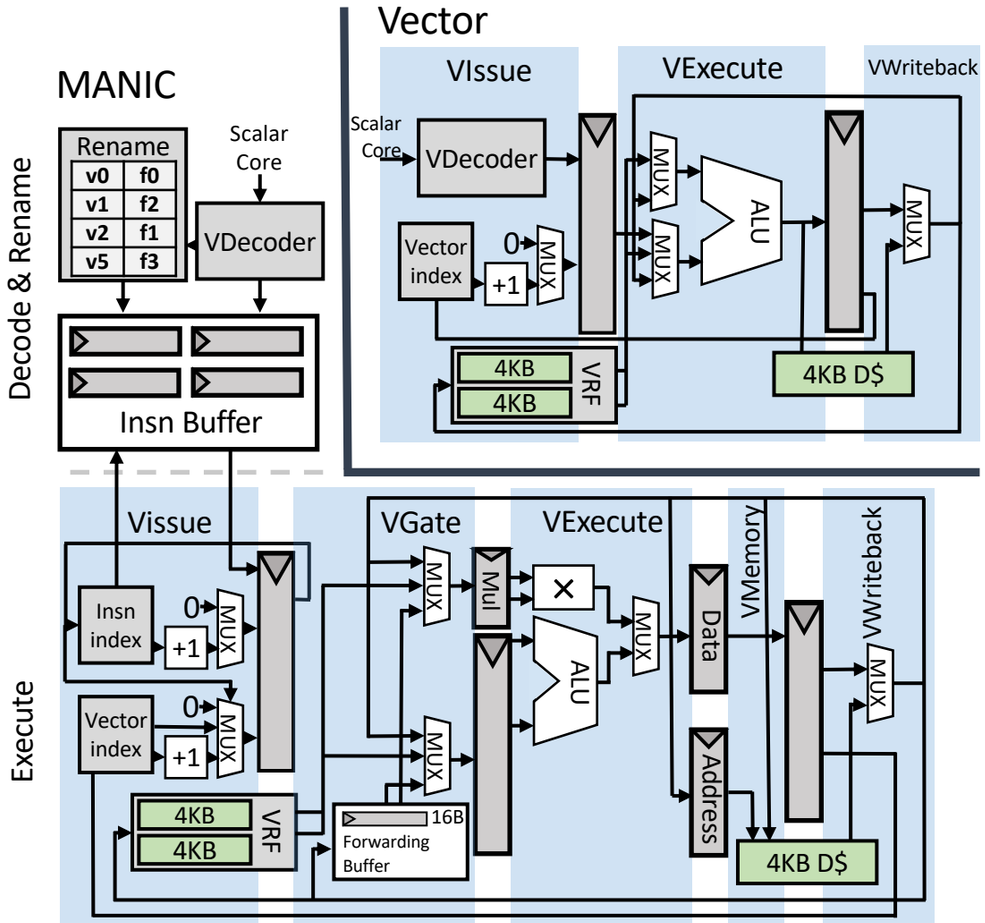

Decode & Rename
Execute
Rename
v0 f0
v1 f2
v2 f1
v5 f3
Scalar Core
VDecoder
Insn Buffer
Vissue
Insn index
Vector index
0
+1
MUX
4KB
4KB
VRF
VGate
16B
Forwarding Buffer
Mul
VExecute
ALU
Data
Address
VMemory
4KB D$
VWriteback

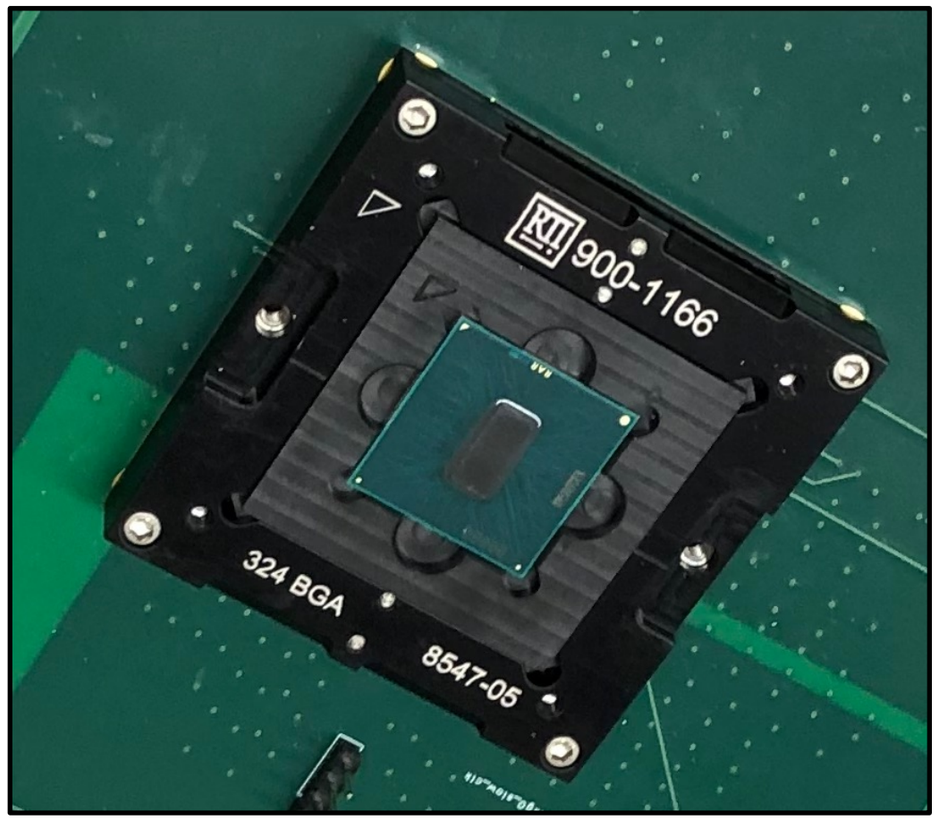

Scalar
MANIC
MRAM
Caches
SRAM

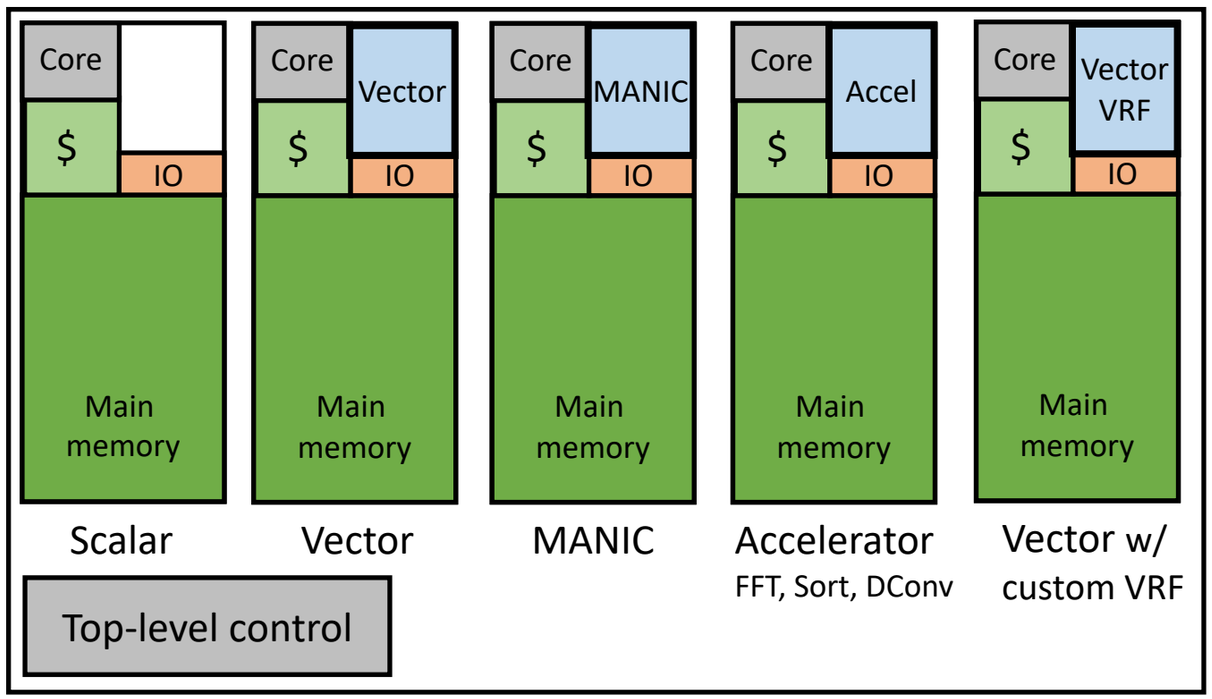

RISC-V
Scalar
Core
Minergy
Arbiter
I2C
GPIO
IO Bus
2KB
I$
4KB D$
4KB
VRF
Arbiter
Boot
ROM
Main
Memory
64KB
SRAM
256KB
MRAM

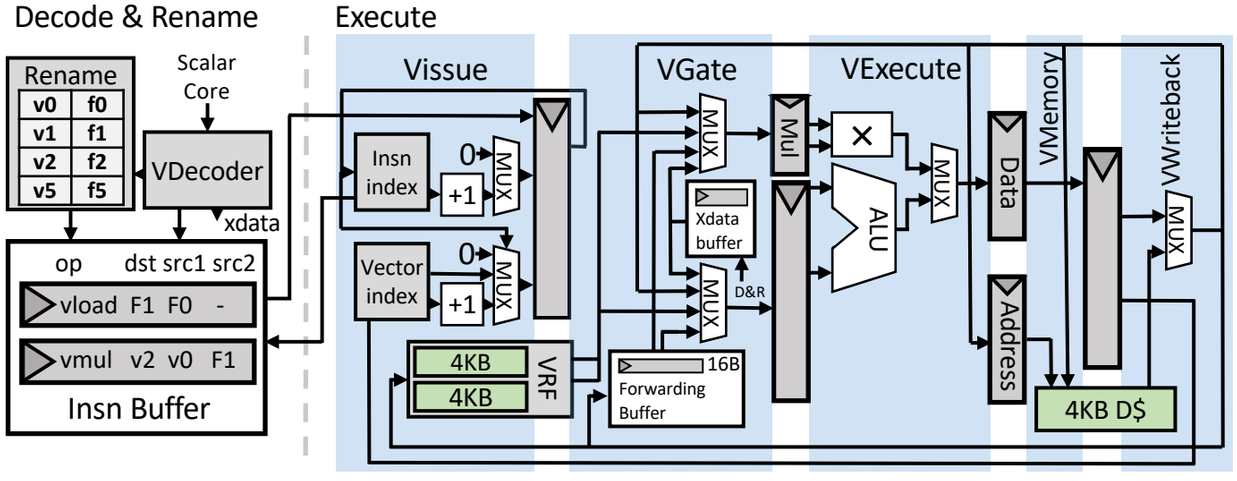

Vector
MANIC
RISC-V Scalar Core
Vector
I2C
GPIO
IO Bus
Arbiter
2KB I$
4KB D$
4KB
4KB
VRF
Arbiter
Boot ROM
Main Memory
64KB SRAM
256KB MRAM
RISC-V Scalar Core
MANIC
I2C
GPIO
IO Bus
Arbiter
2KB I$
4KB D$
4KB VRF
Arbiter
Boot ROM
Main Memory
64KB SRAM
256KB MRAM

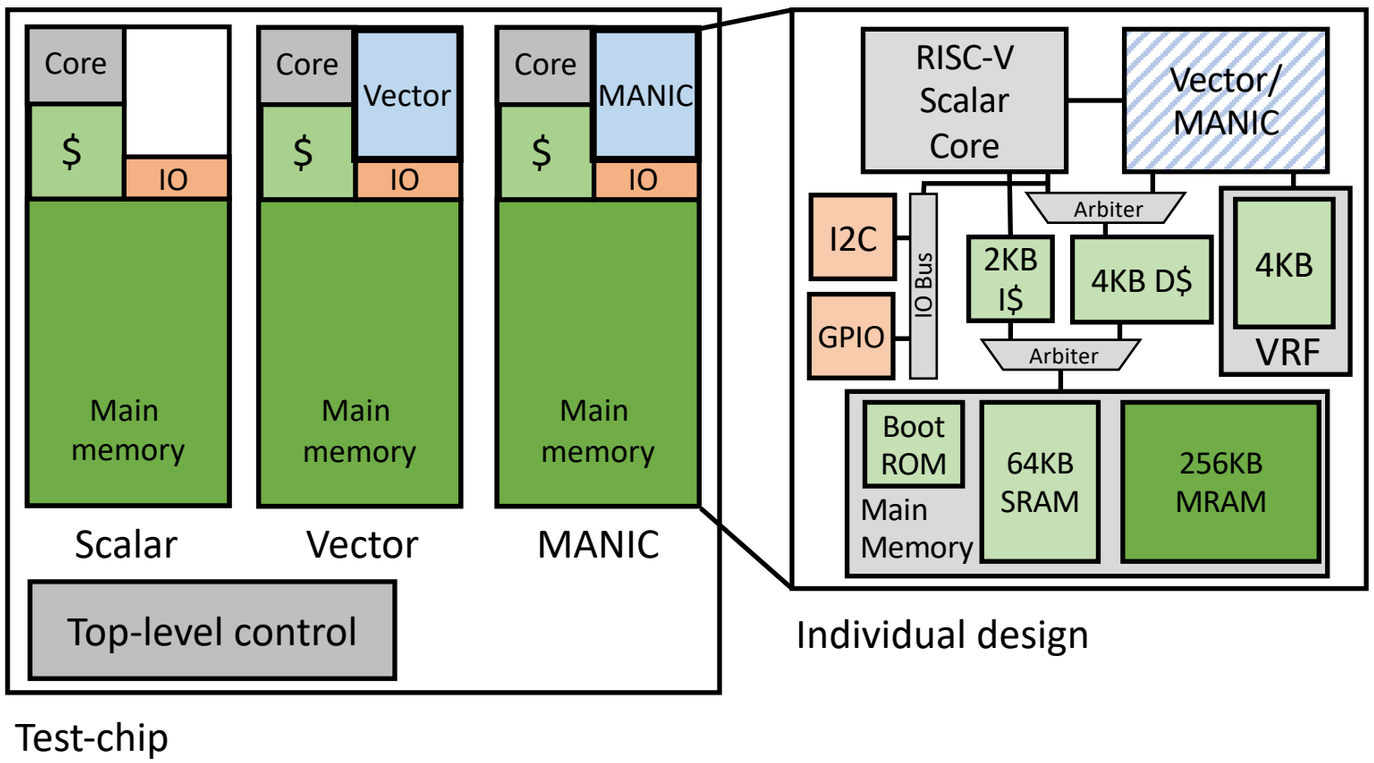

Voltage (V)
0.7
0.6
0.5
0.4
3.6
4.5
6.1
7.2
9.1
12.2
14.4
18.1
24.5
28.8
36.3
48.9
Frequency (MHz)


MANIC
Decode & Rename
Rename
v0 f0
v1 f2
v2 f1
v5 f3
Scalar Core
VDecoder
Insn Buffer
Vector
VIssue
VExecute
VWriteback
Scalar Core
VDecoder
Vector index
0
+1
MUX
ALU
4KB
4KB
VRF
4KB D$
Execute
Vissue
VGate
VExecute
VMemory
VWriteback
Insn index
Vector index
Mul
Data
Address
16B
Forwarding Buffer
4KB D$


900-1166
324 BGA
8547-05


Core
$
IO
Main memory
Scalar
Core
Vector
$
IO
Main memory
Vector
Core
MANIC
$
IO
Main memory
MANIC
Core
Accel
$
IO
Main memory
Accelerator
FFT, Sort, DConv
Core
Vector VRF
$
IO
Main memory
Vector w/ custom VRF
Top-level control


Decode & Rename
Execute
Rename
v0 f0
v1 f1
v2 f2
v5 f5
Scalar Core
VDecoder
xdata
op dst src1 src2
vload F1 F0 -
vmul v2 v0 F1
Insn Buffer
Vissue
Insn index
0
+1
MUX
Vector index
4KB
4KB
VRF
VGate
Xdata buffer
D&R
16B
Forwarding Buffer
Mul
VExecute
ALU
Data
Address
VMemory
4KB D$
VWriteback


Core
$
IO
Main memory
Scalar
Core
Vector
$
IO
Main memory
Vector
Core
MANIC
$
IO
Main memory
MANIC
Top-level control
RISC-V Scalar Core
Vector/ MANIC
Arbiter
I2C
GPIO
IO Bus
2KB I$
4KB D$
4KB
VRF
Arbiter
Boot ROM
Main Memory
64KB SRAM
256KB MRAM
Individual design

Test-chip