Improving Energy Efficiency of Data Irregular Codes in a Vector-Thread Architecture

Alex Bishara, Richard Xia

Overview

The vector-thread (VT) architecture can perform poorly on certain data-irregular codes.

We examined two hardware additions that significantly improve performance on certain classes of irregular code.

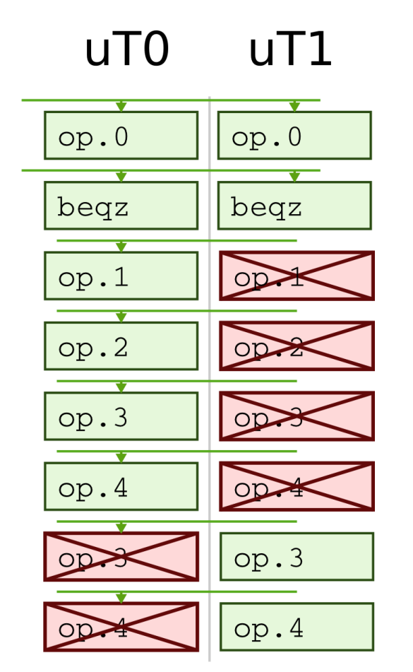
Vector Fragment Reconvergence

When vector fragments diverge, they stay divergent until the end of the vector fetch block.

This results in poor performance in common control sequences such as `if` statements within loops.

```
00: loop:
04:    op.0
08:    beqz skip
0c:    op.1
10:    op.2
14: skip:
18:    op.3
1c:    op.4
20:    j loop
```

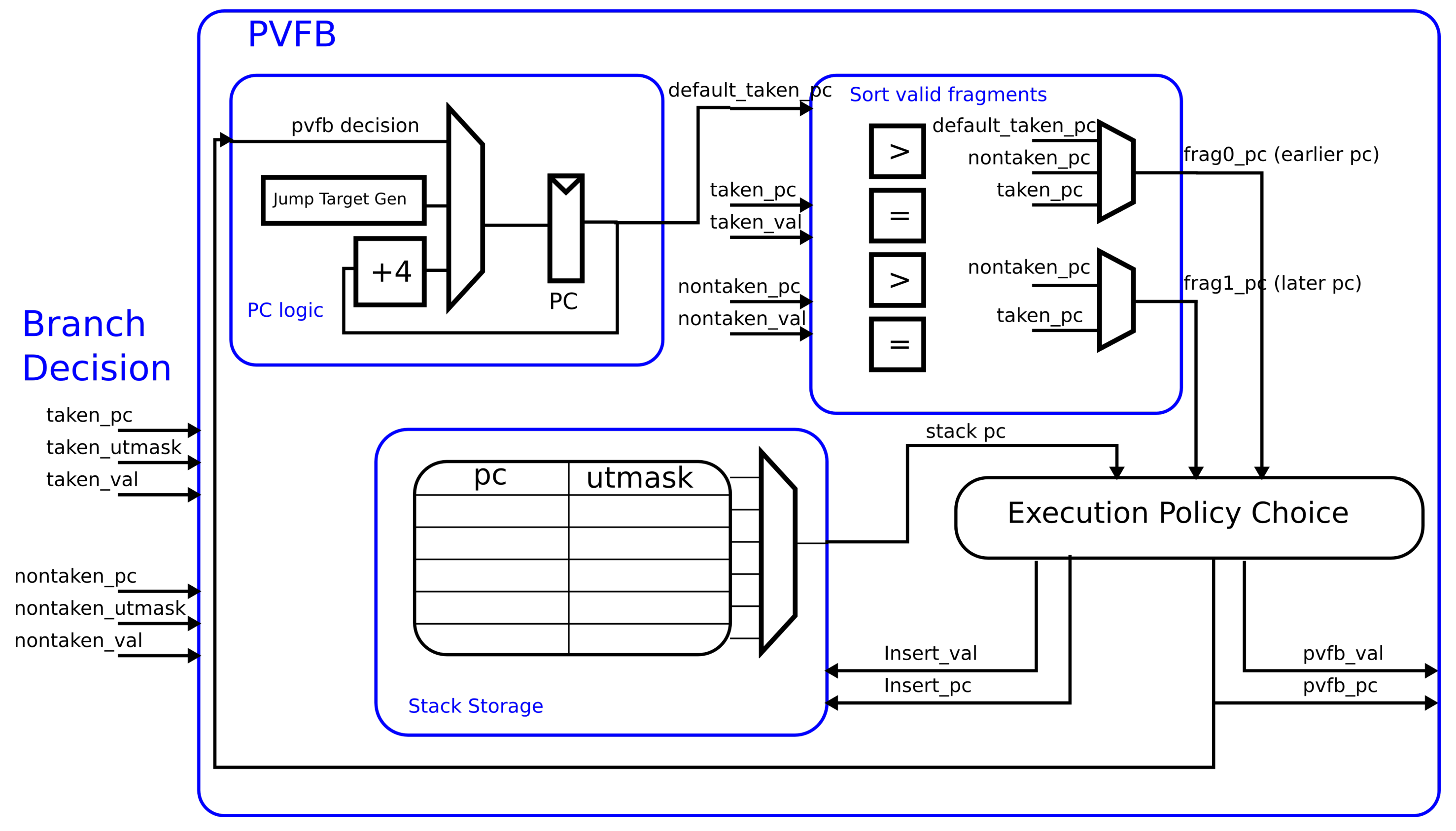
(a) sample code



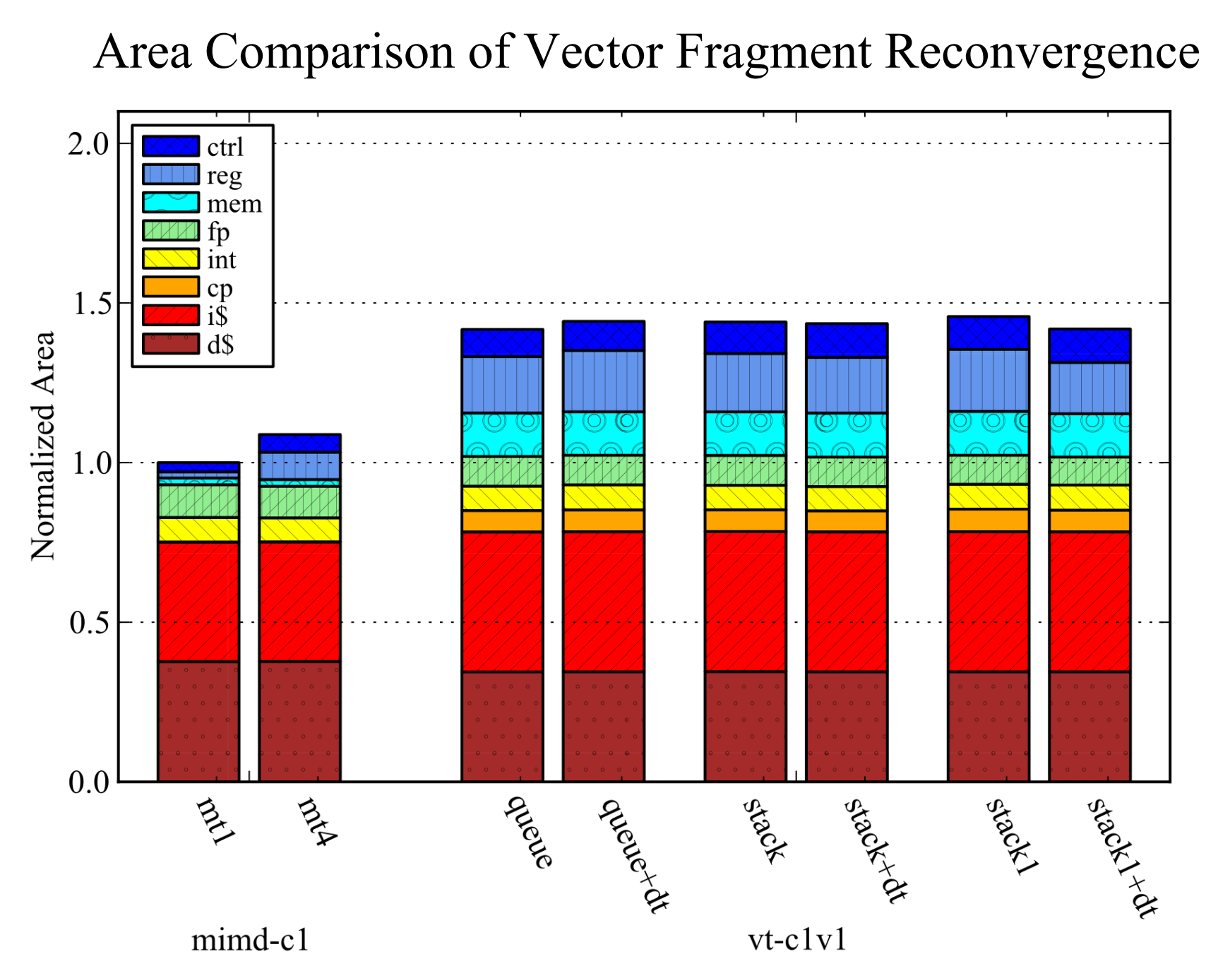
(b) execution diagram

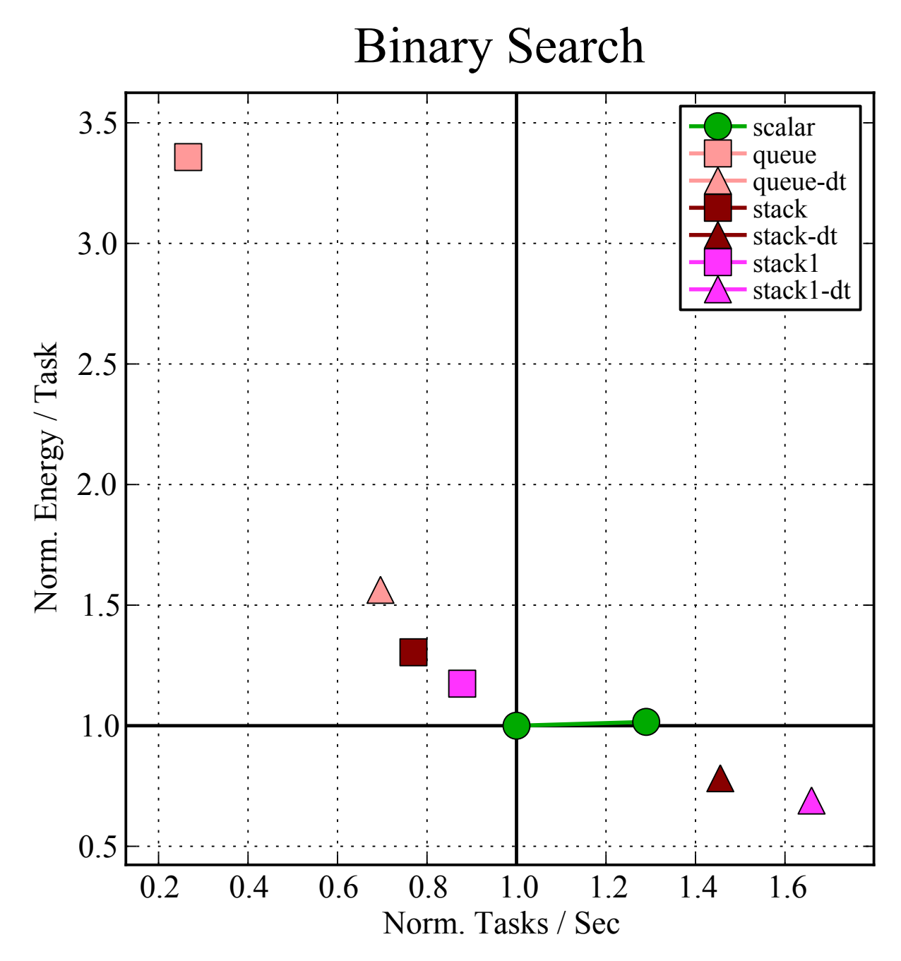
The pending vector reconvergence buffer (PVFB) stores and retrieves the PC and microthread masks of each fragment. The vector unit can push and pop from the PVFB, which normally uses a FIFO policy.

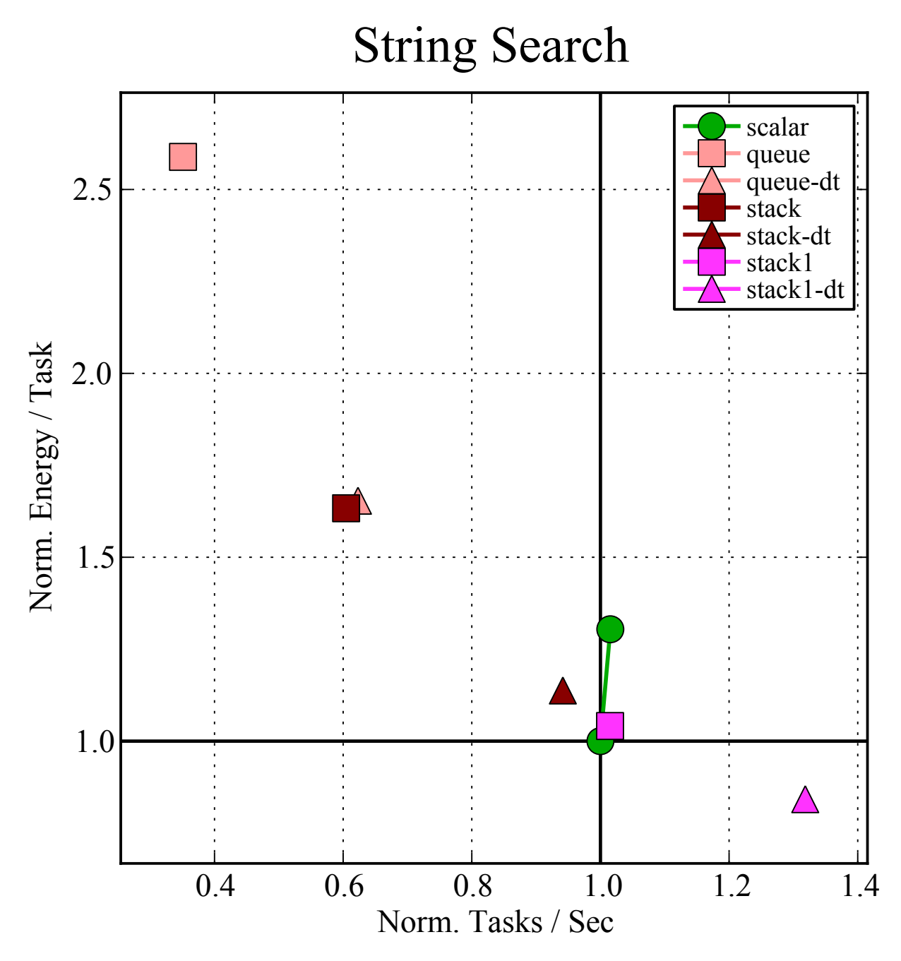
We implemented two variations of a special hardware stack to allow inserted fragments to reconverge if PCs match.



Although the special stacks are much more complex to implement, they incur no additional area cost and greatly improve the performance of divergent codes.







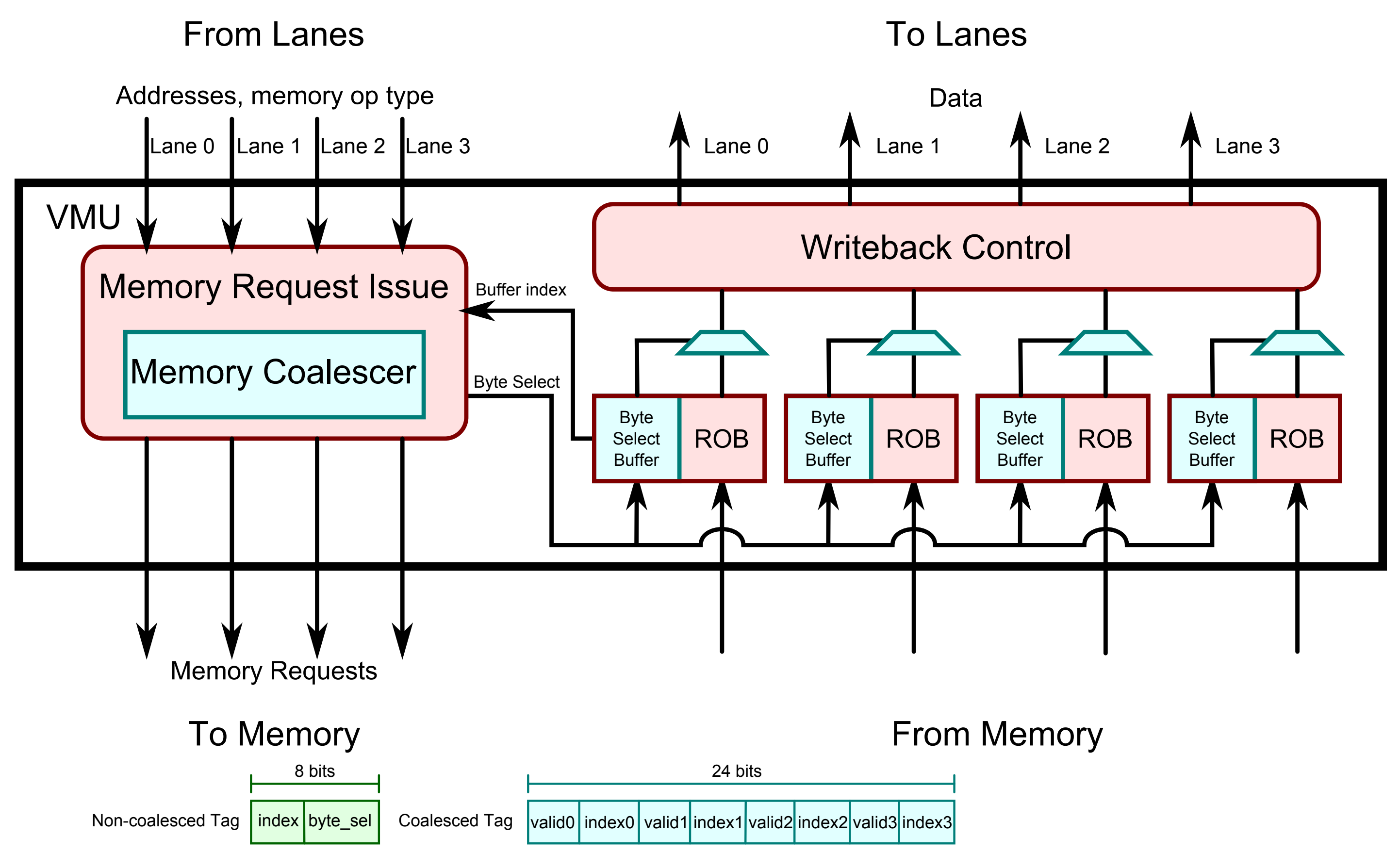
Previously, irregular codes would perform much worse on a VT machine than a comparable scalar machine. Our implementation allows VT to run these codes even better than scalar machines.

Memory Coalescer

When using microthread loads and stores, cache conflicts can degrade performance on certain common codes.

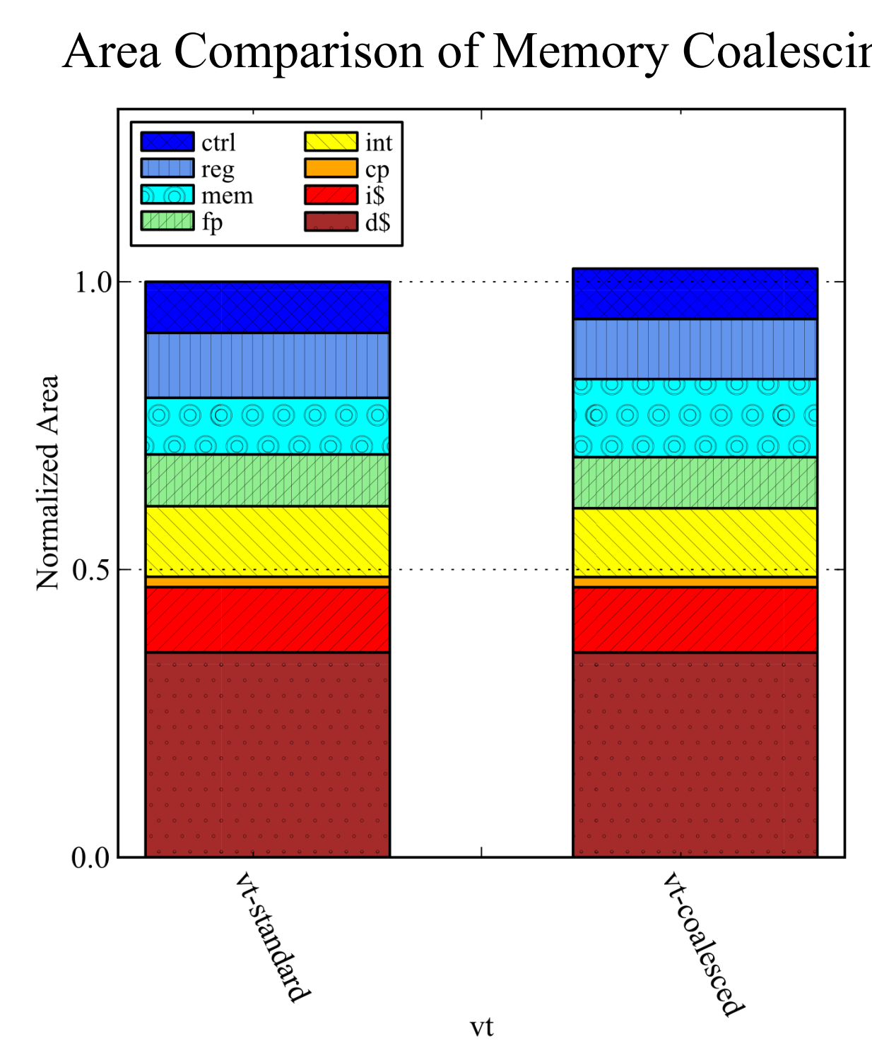
Sequential memory accesses of small strides can have cache conflicts so badly that they allow only one memory access per cycle.

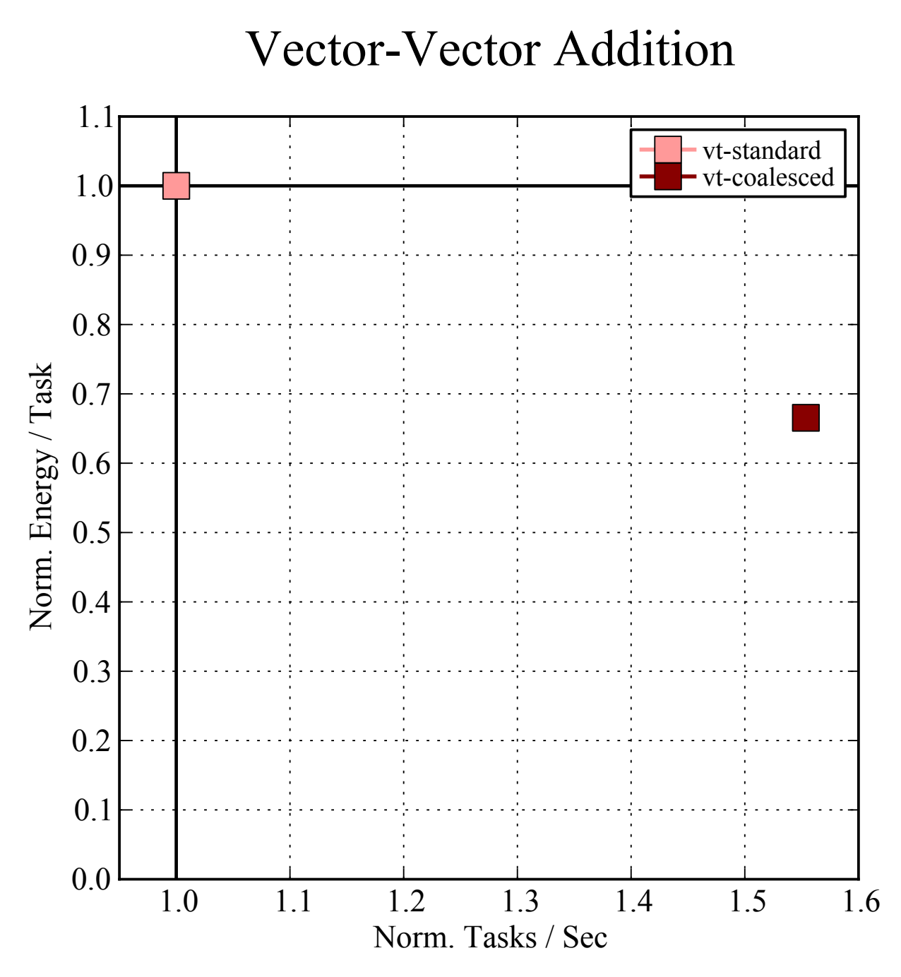
Our memory coalescer allows a single memory request to satisfy memory requests of multiple lanes simultaneously. The vector memory unit dynamically compares memory addresses and stores word/halfword/byte select information in an extra buffer.

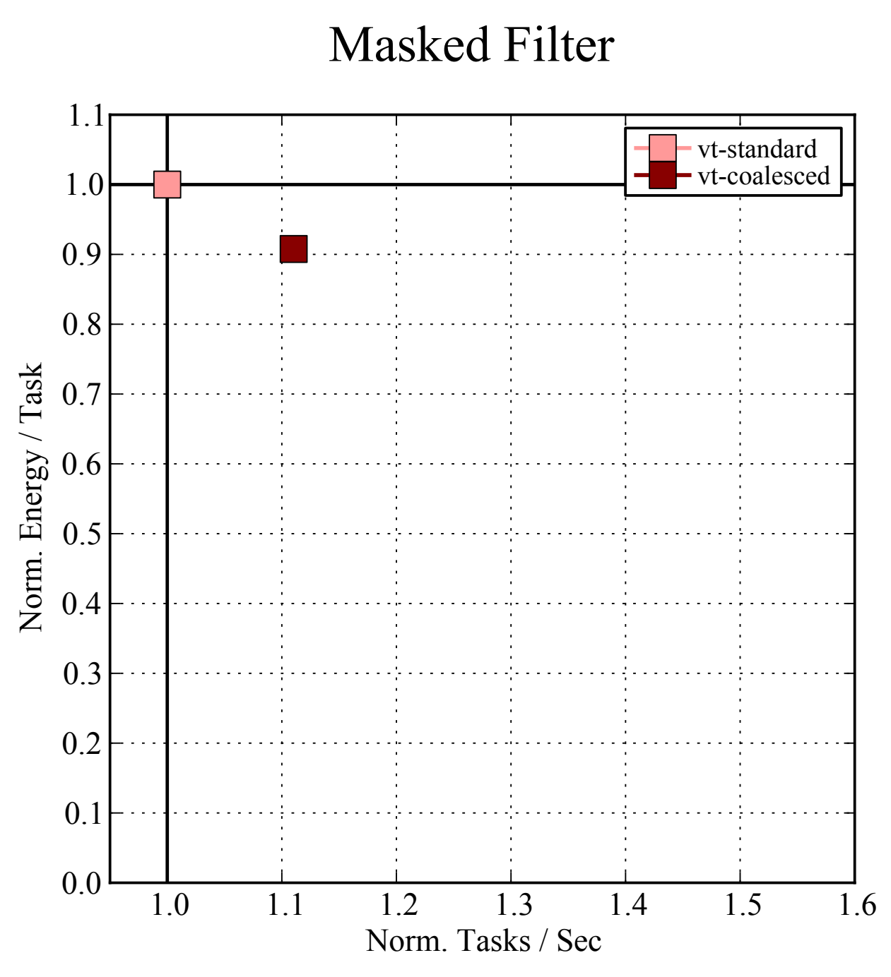


The changes to the vector unit add an insignificant amount of area.

By removing cache conflicts, performance can improve by as much as 55%, depending on the severity of the original cache conflicts.


Area Comparison of Memory Coalescing
Normalized Area
ctrl
reg
mem
fp
int
cp
i$
d$
vt-standard
vt-coalesced
vt


Vector-Vector Addition
Norm. Energy / Task
Norm. Tasks / Sec
vt-standard
vt-coalesced


Masked Filter
Norm. Energy / Task
Norm. Tasks / Sec
vt-standard
vt-coalesced