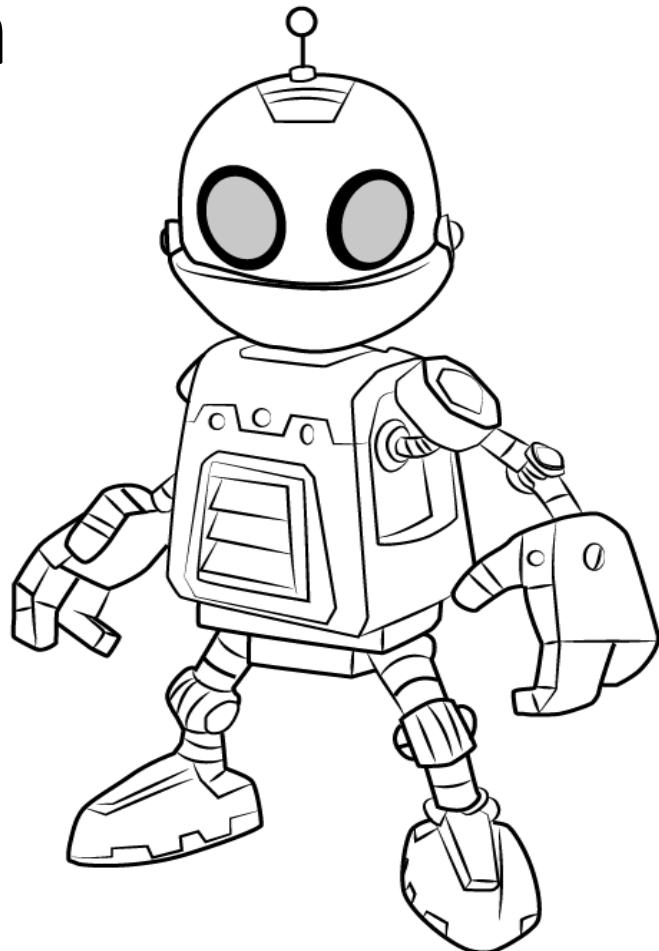


Clank: Architectural support for intermittent computation

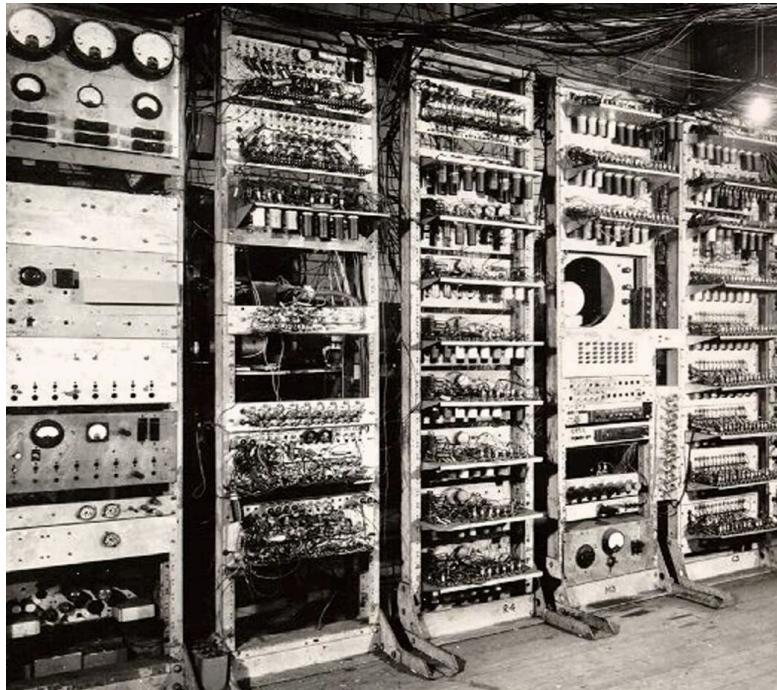
Matthew Hicks

Virginia Tech

mdhicks2@VT.edu



From whole-room computers to millimeter-scale computers



ENIAC 1946.



Millimeter-Scale Computers: Now With Deep-Learning Neural Networks on Board,
Katherine Bourzac, IEEE Spectrum, Feb. 2017.²

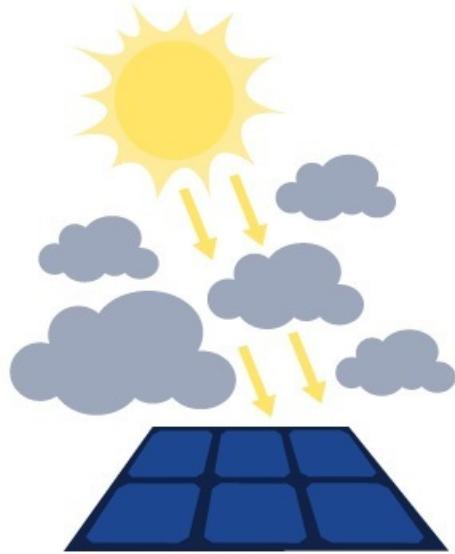
Batteries are the barrier to innovation



Harvested energy is the solution



Making correct forward progress with frequent power cycles is challenging



Unpredictable power

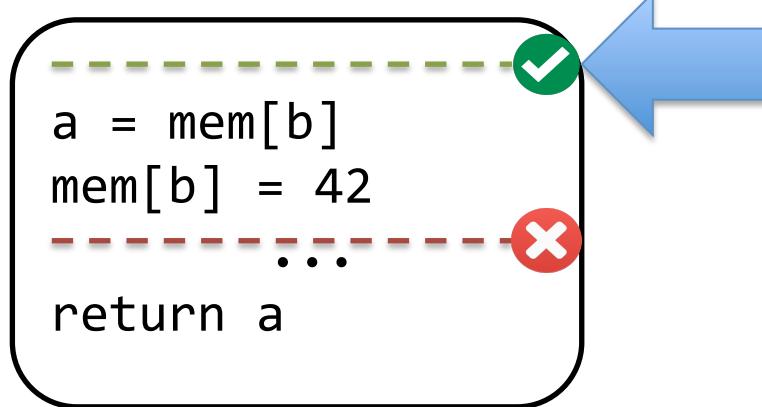


Never enough power



Mixed volatility memory

Re-executions and mixed-volatility cause incorrectness



	Initial value	Correct execution	Value at reset	Error
a	x	y	x	42
mem[b]	y	42	42	42

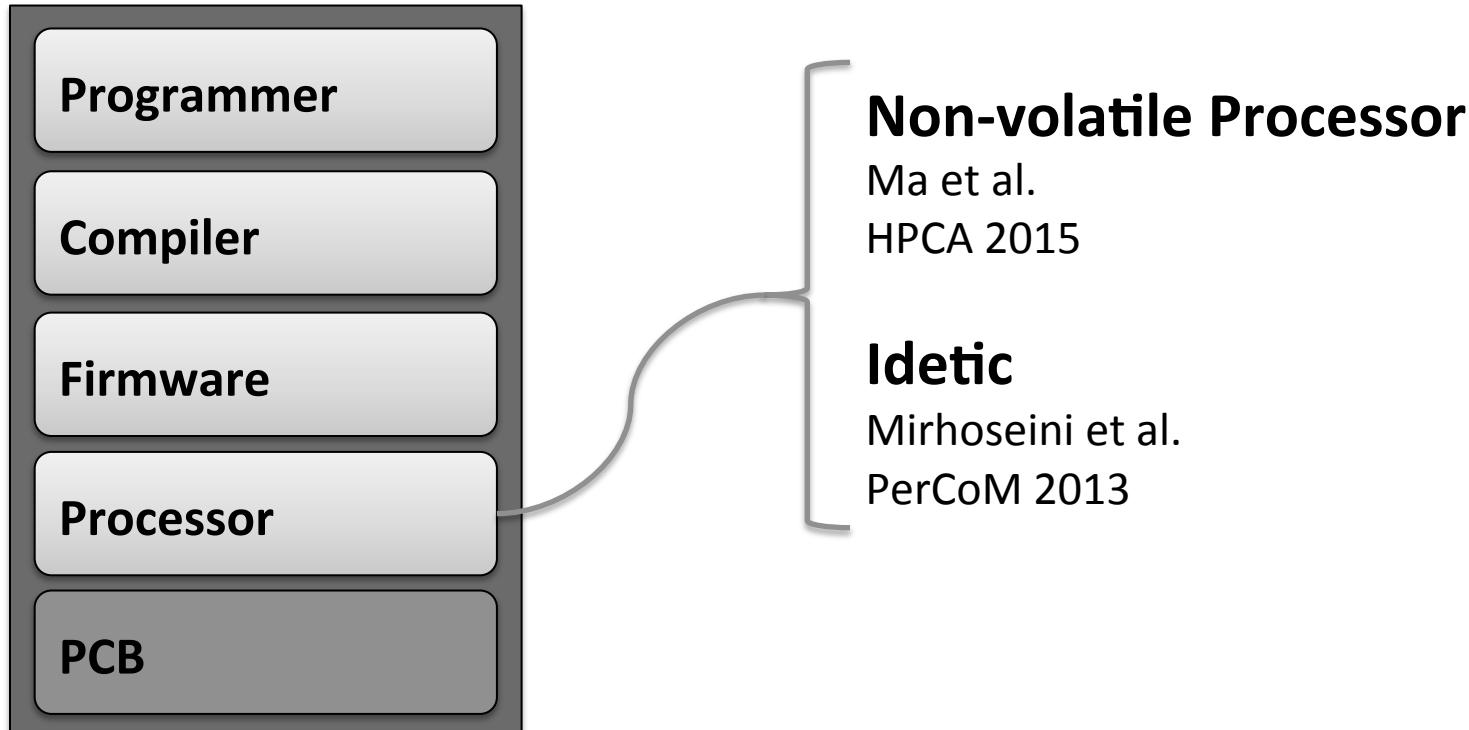
Write after Read (WAR)**

Voltage monitoring steals energy from software

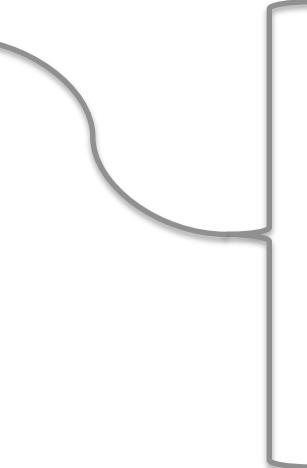


- Mementos**
Ransford et al.
ASPLOS 2011
- Hibernus**
Balsamo et al.
Embedded Systems Letters 2014
- QUICKRECALL**
Jayakumar et al.
Embedded Systems and VLSI Design 2014

Heavy weight hardware modification is expensive and restrictive



Compilers must operate statically and pessimistically



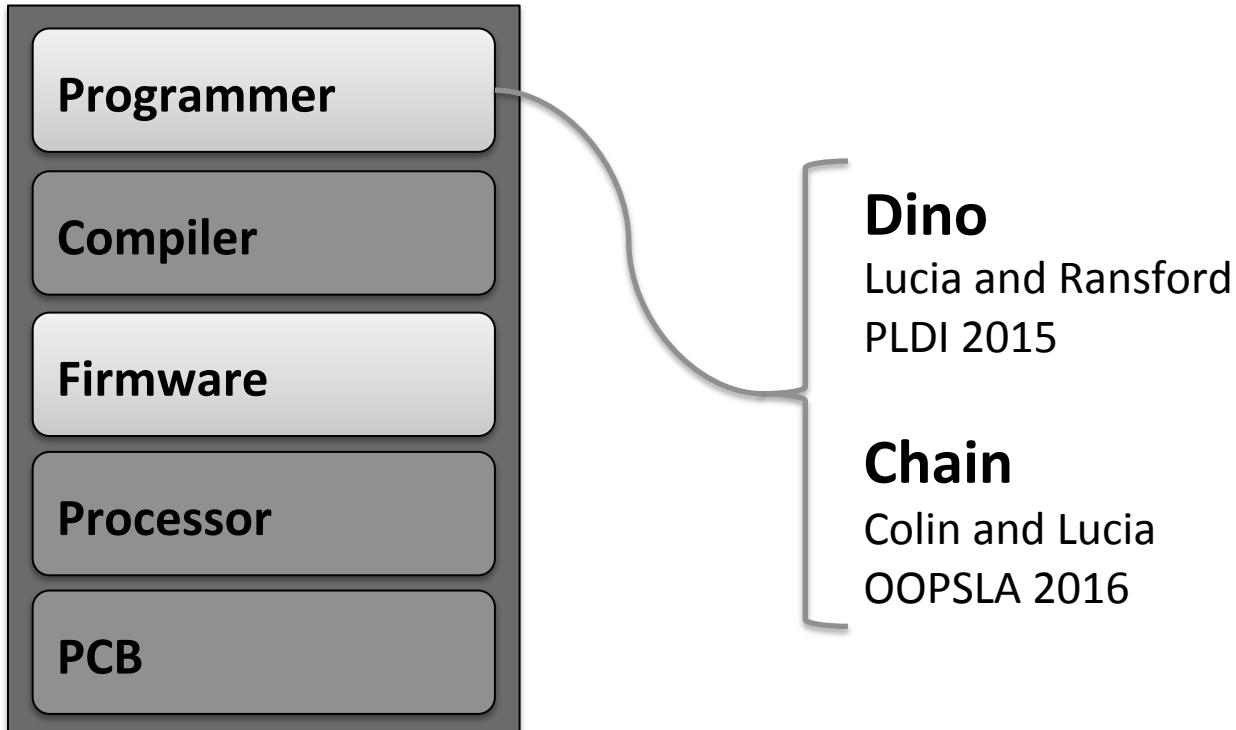
Mementos

Ransford et al.
ASPLOS 2011

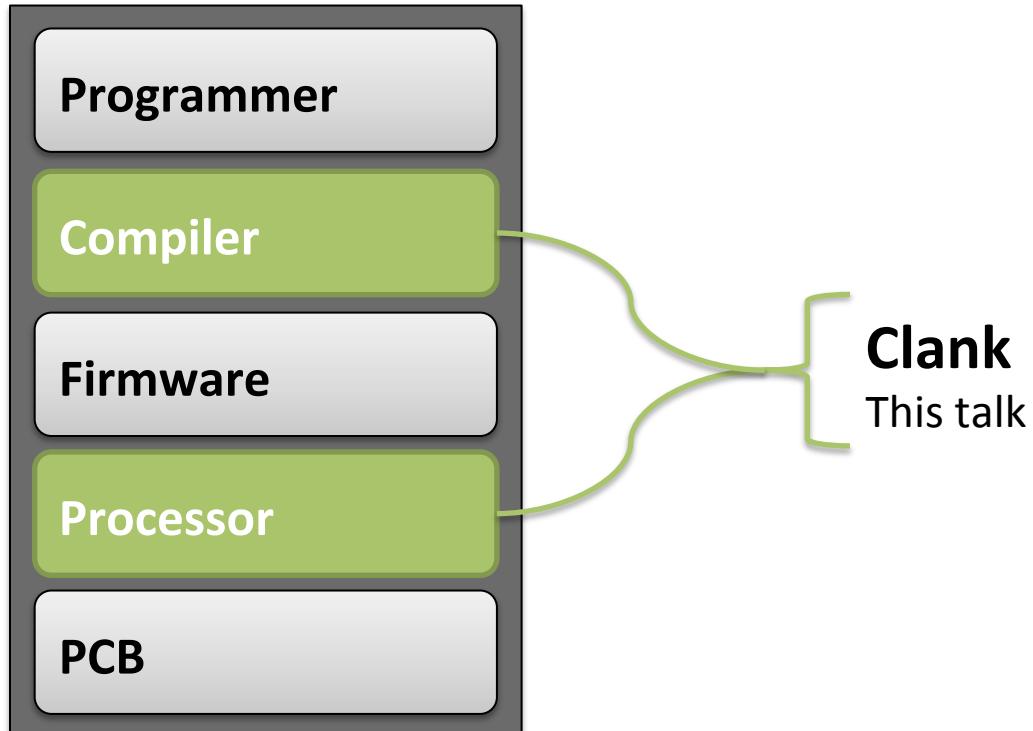
Ratchet

Van der Woude and Hicks
OSDI 2016

Requiring programmer intervention is error prone and limiting



Clank: lightweight hardware modification with minimal compiler support



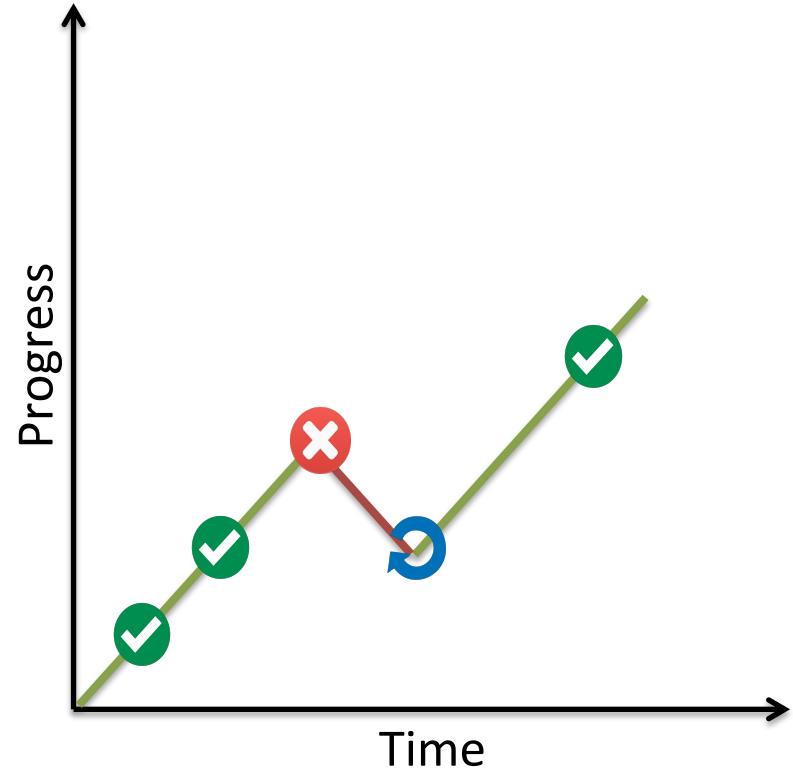
Clank is hardware-level monitoring of a
program's **dynamic idempotence**

Idempotence is about preserving values read for consistent replay

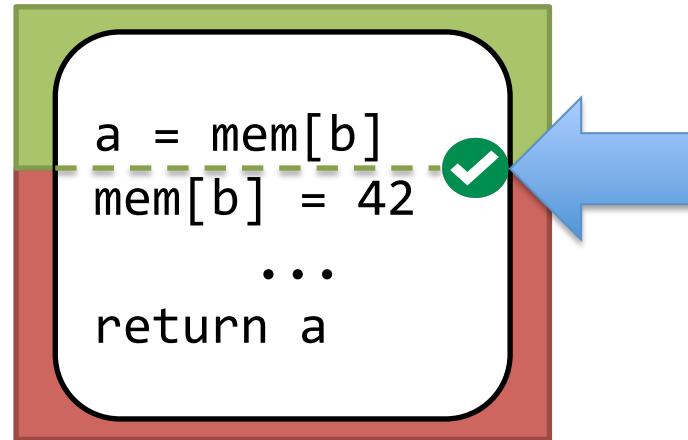
- Guarantees semantically correct re-execution
- It is possible to decompose programs into checkpoint-connected idempotent sections

Checkpoints break idempotence violations

- Save volatile state to non-volatile memory
- After a power fail
 - restart
 - restore
 - continue



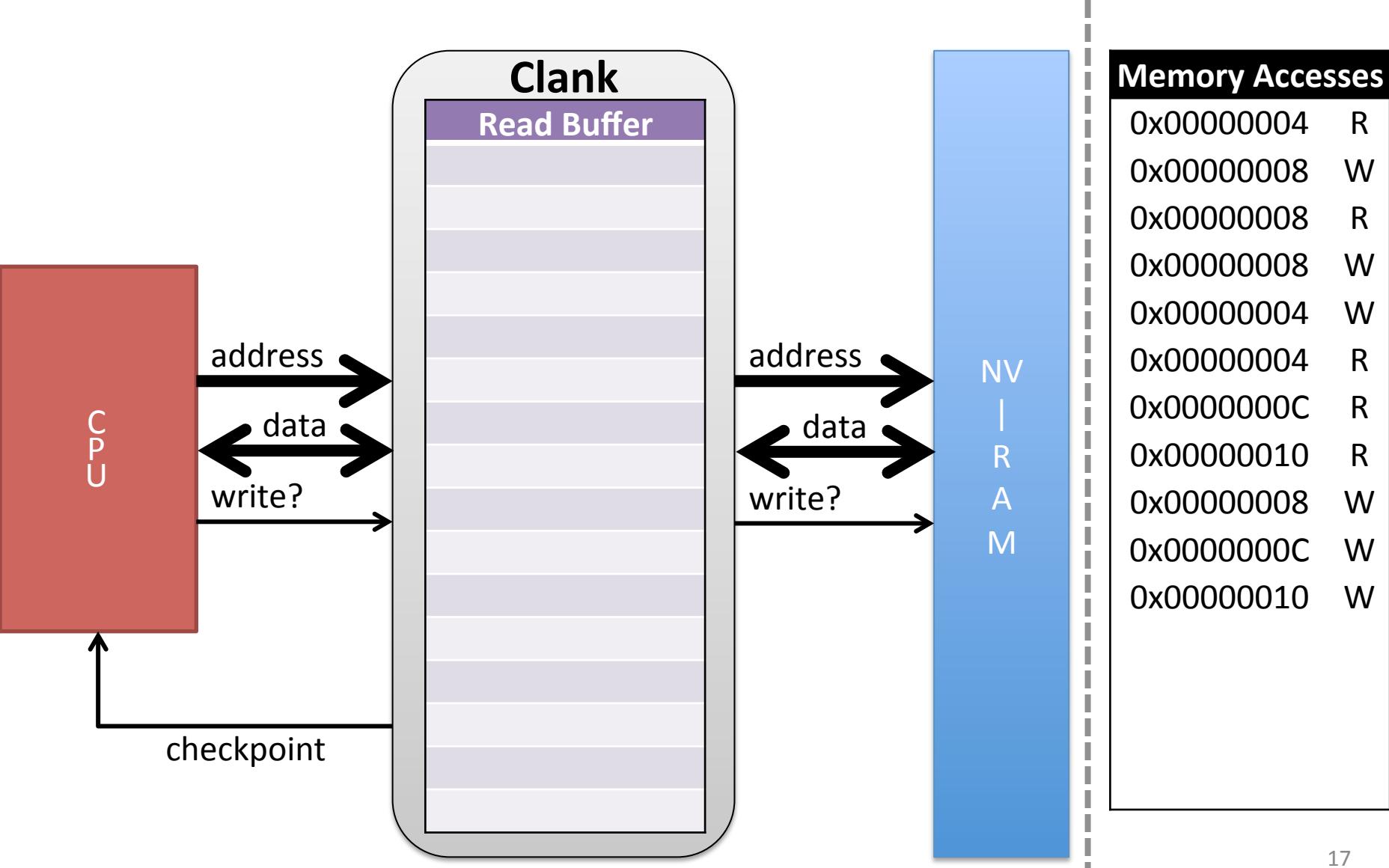
Preserving idempotence enables correct re-execution

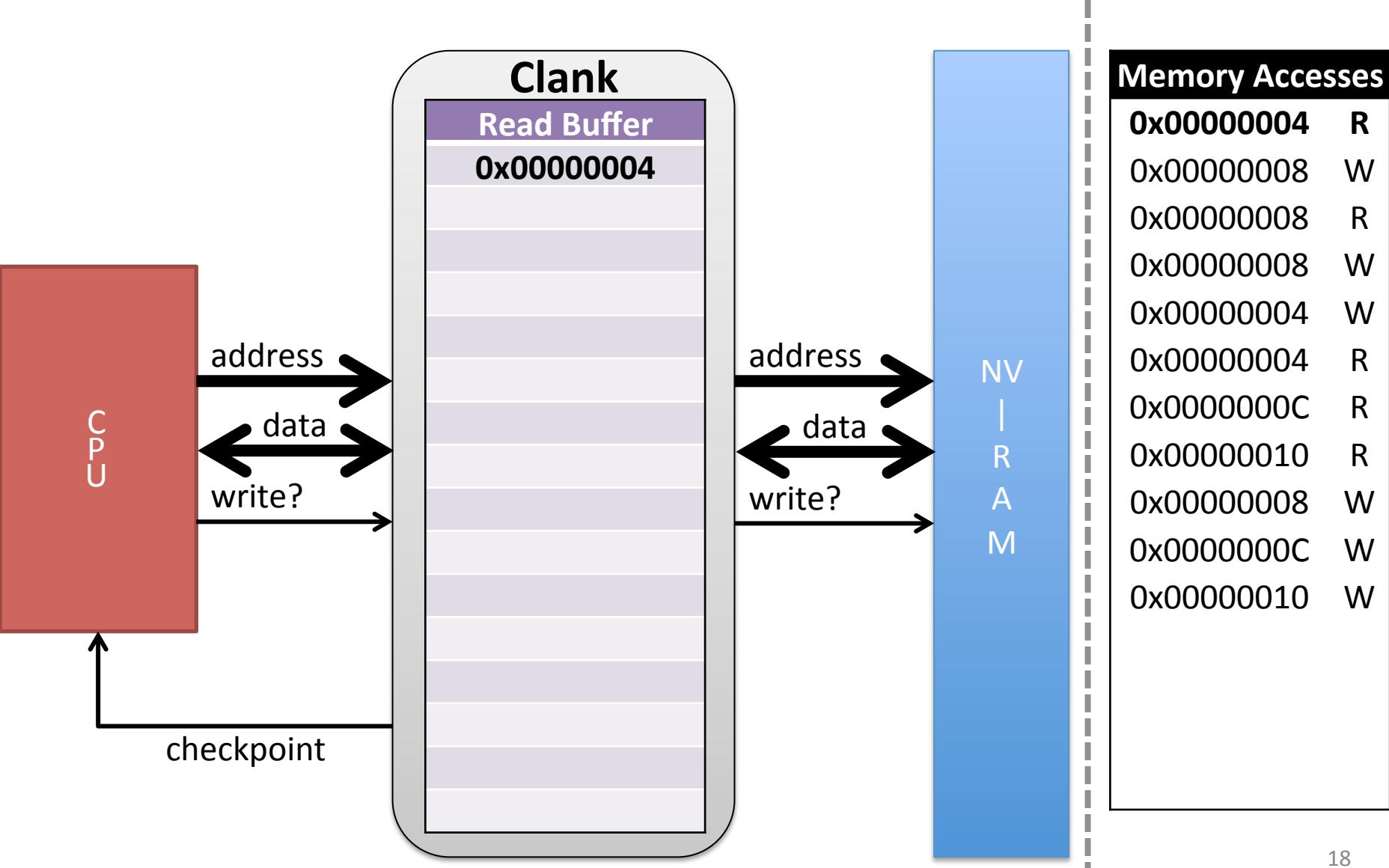


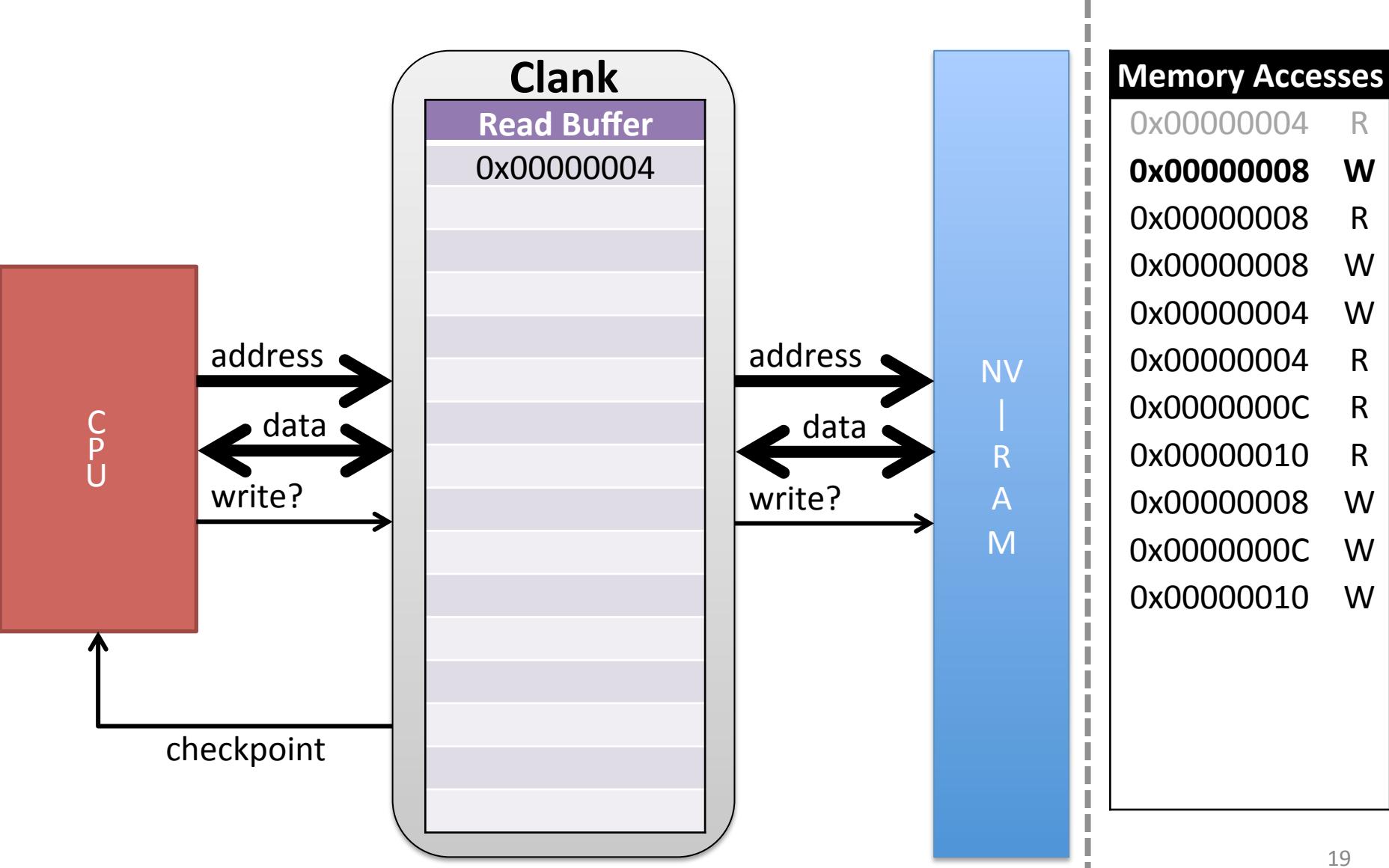
	Initial value	Correct execution	Value at reset	No Error!
a	x	y	y	y
mem[b]	y	42	42	42

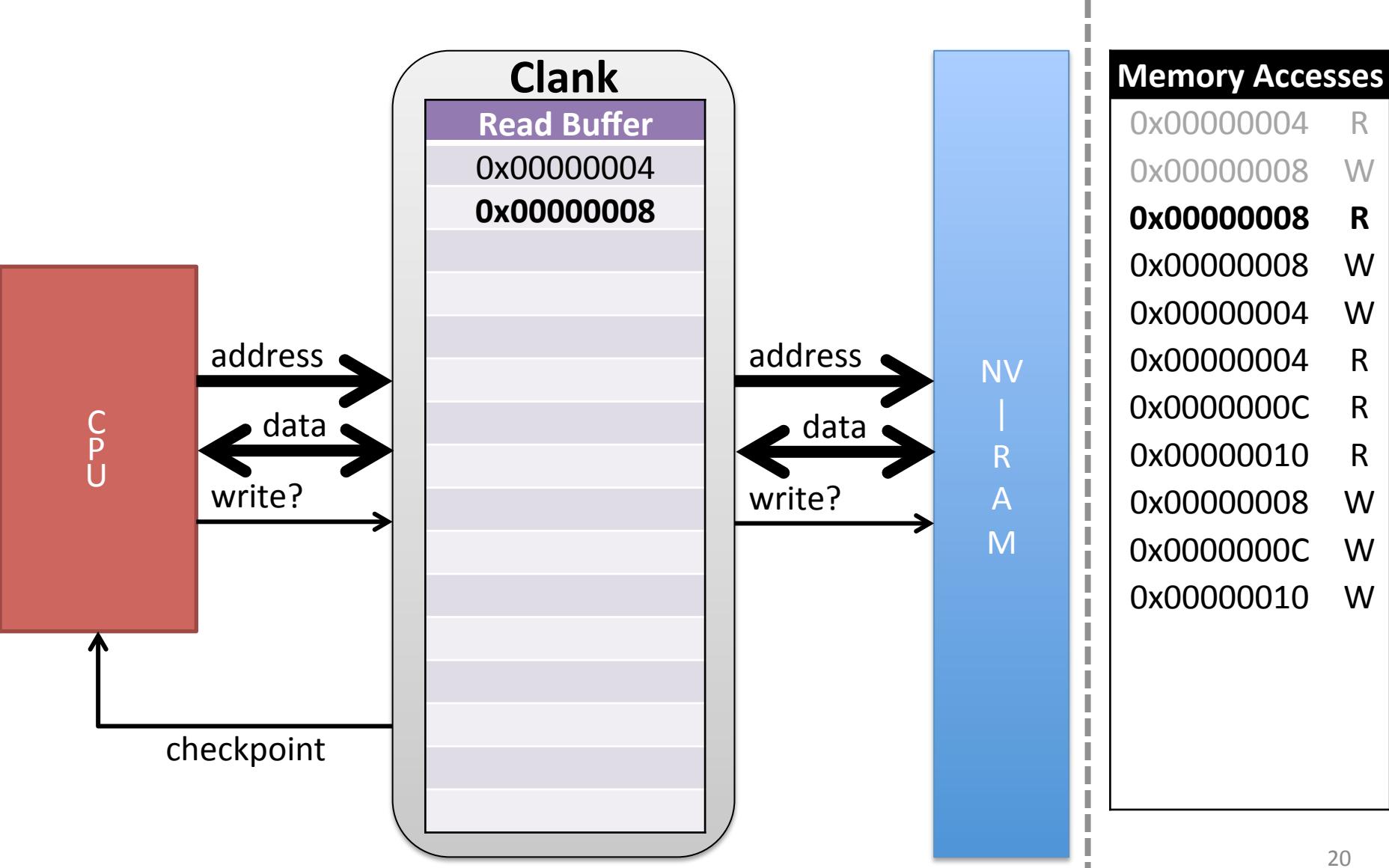
Constructing Clank through design space exploration

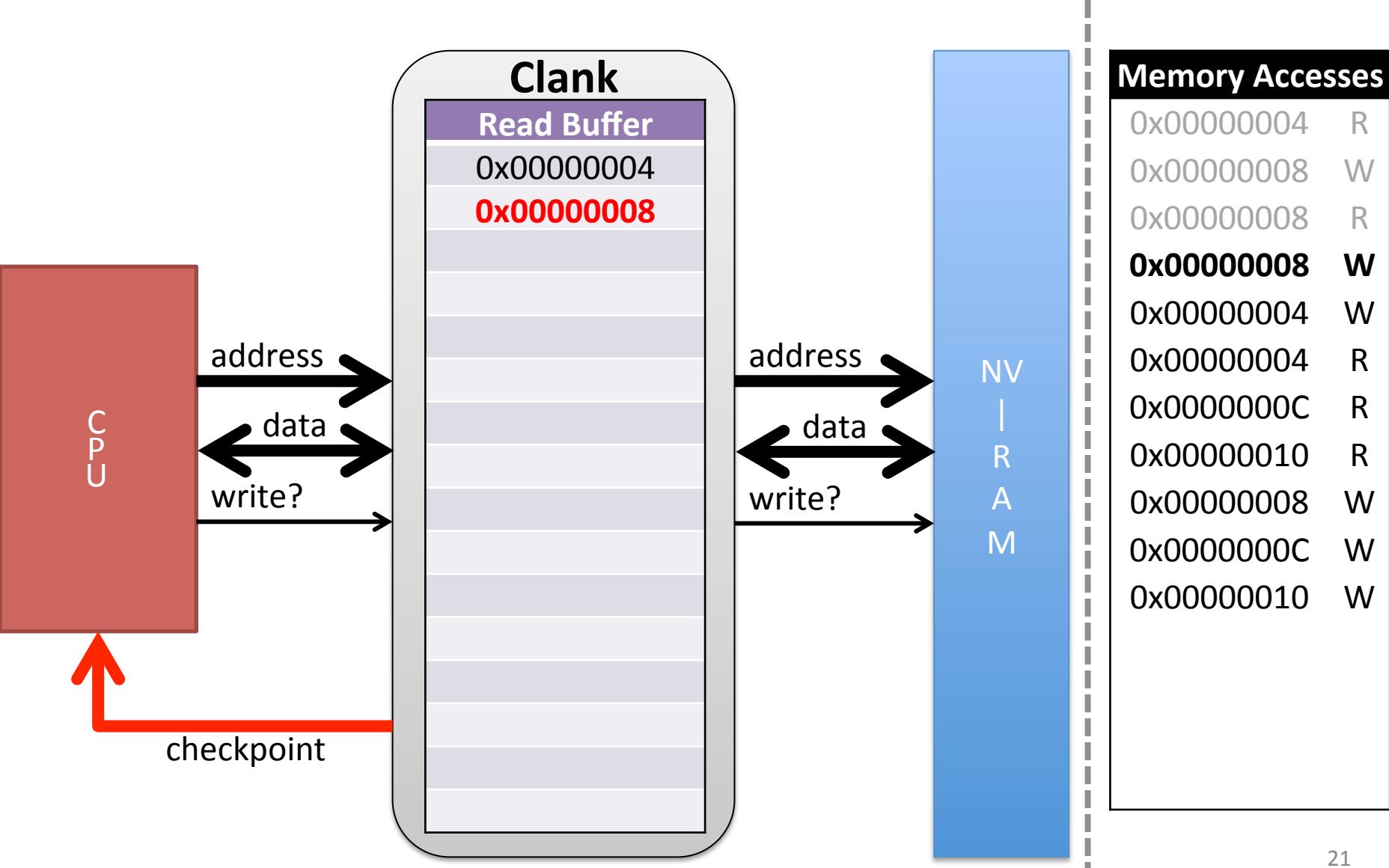
- 23 benchmarks ported from MiBench suite
- Almost 1,000,000 configurations explored per benchmark

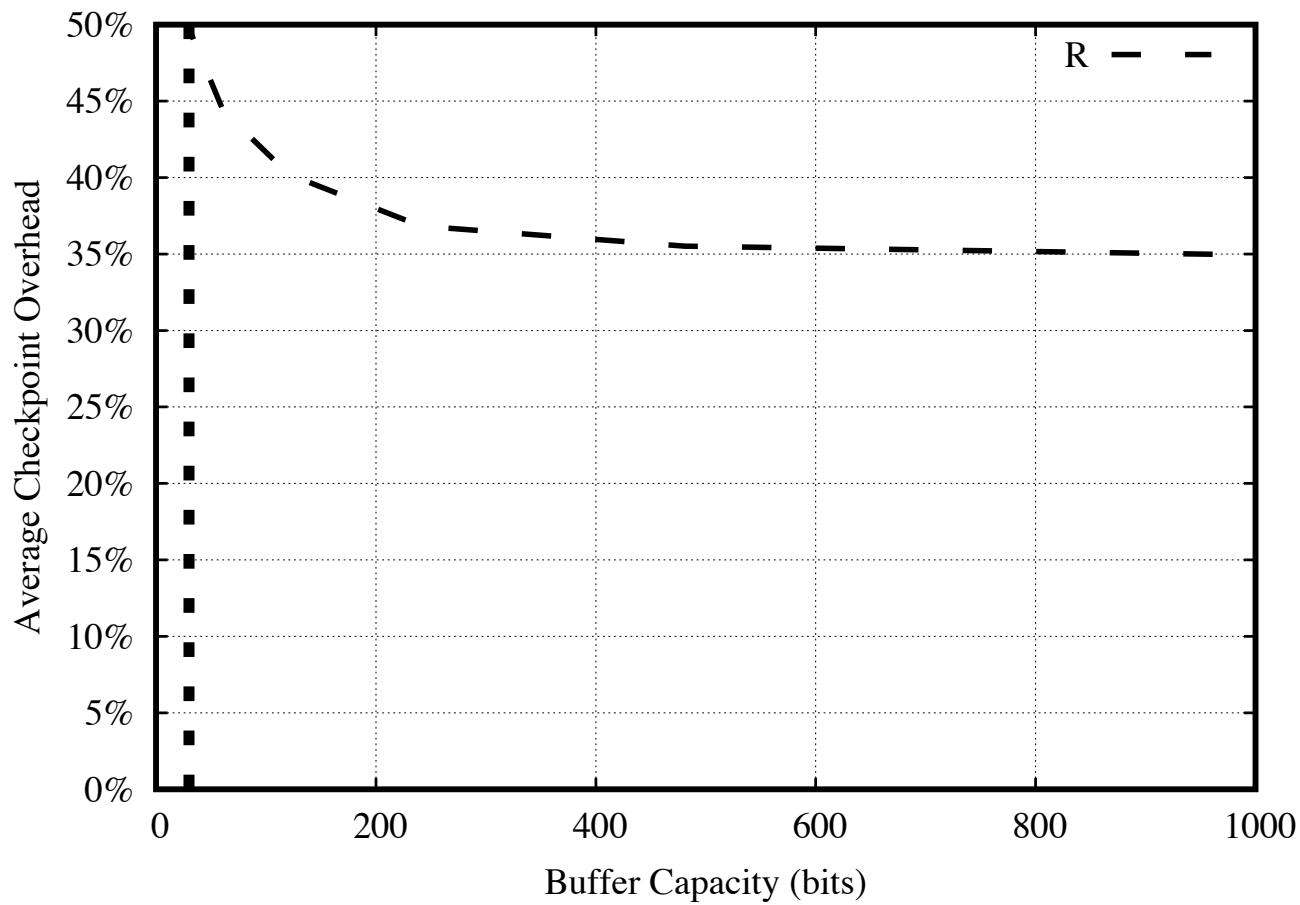




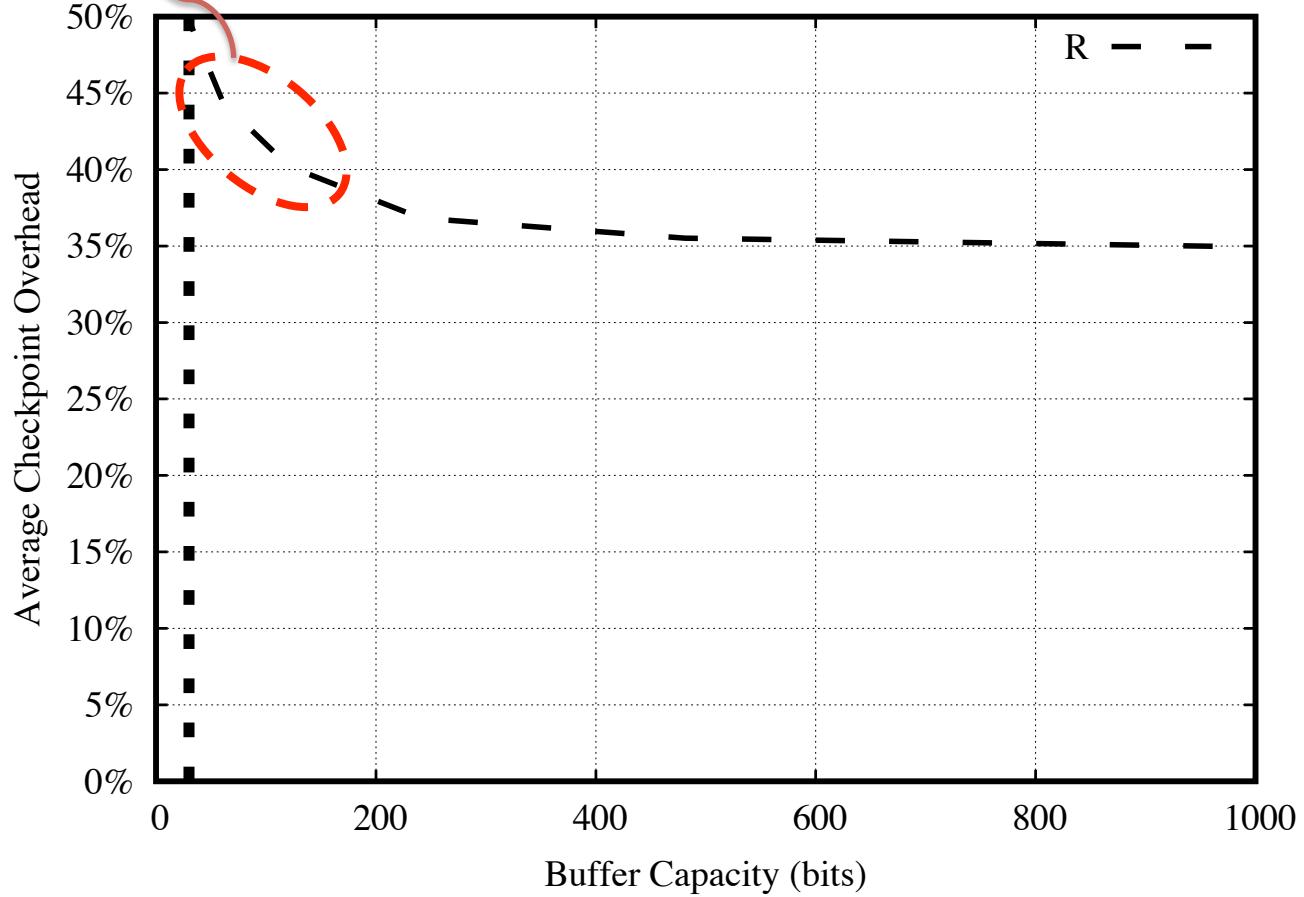




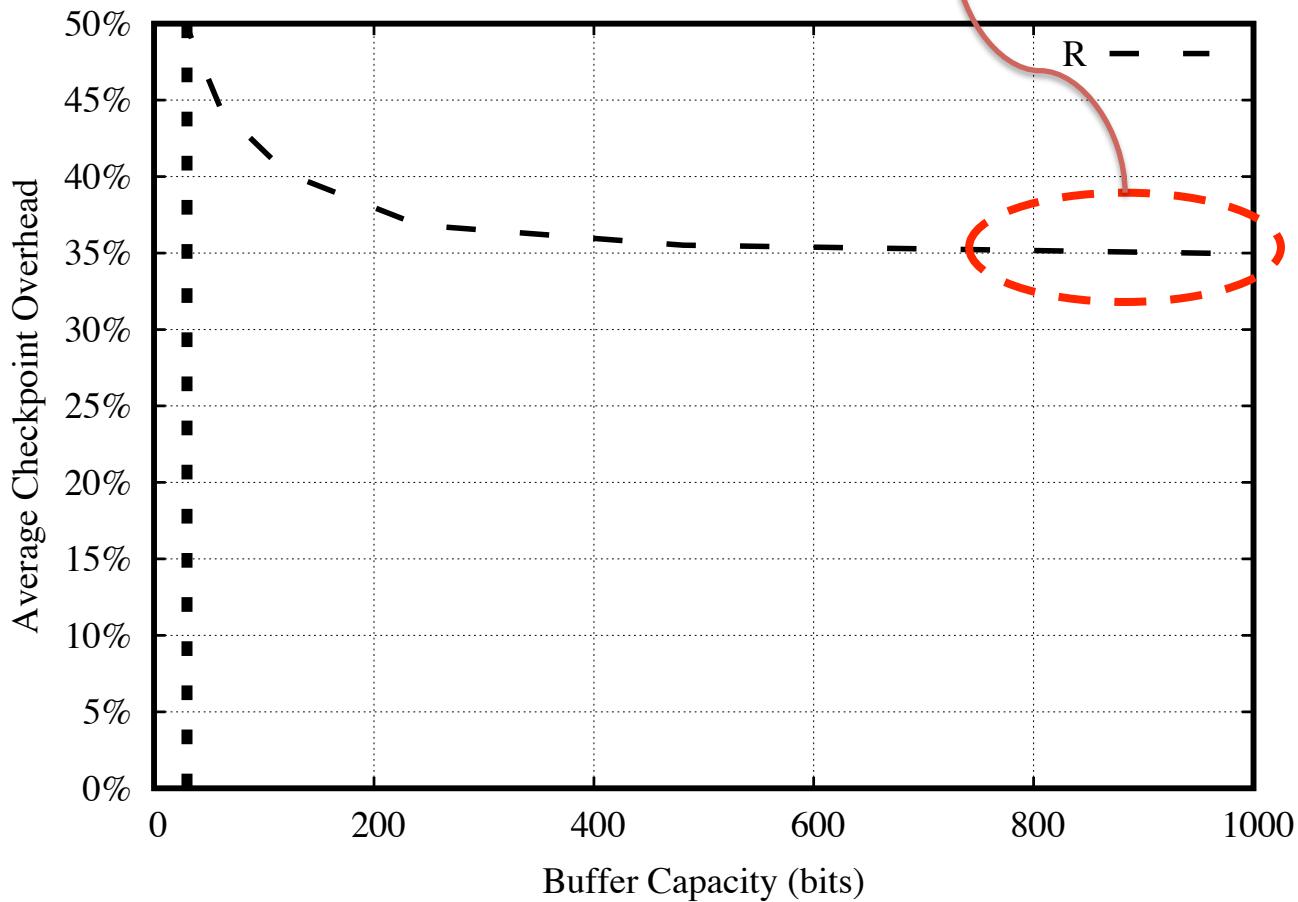


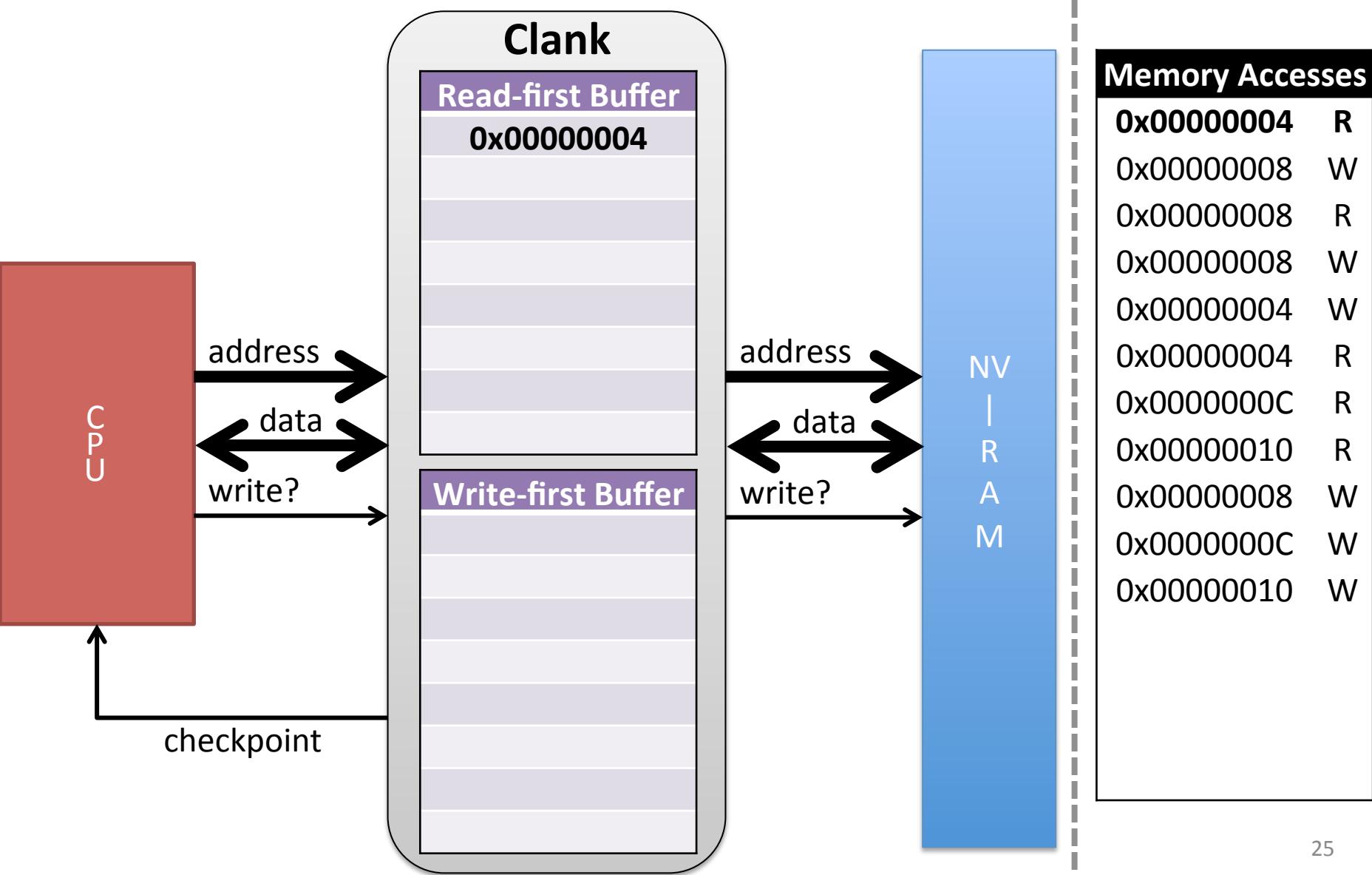


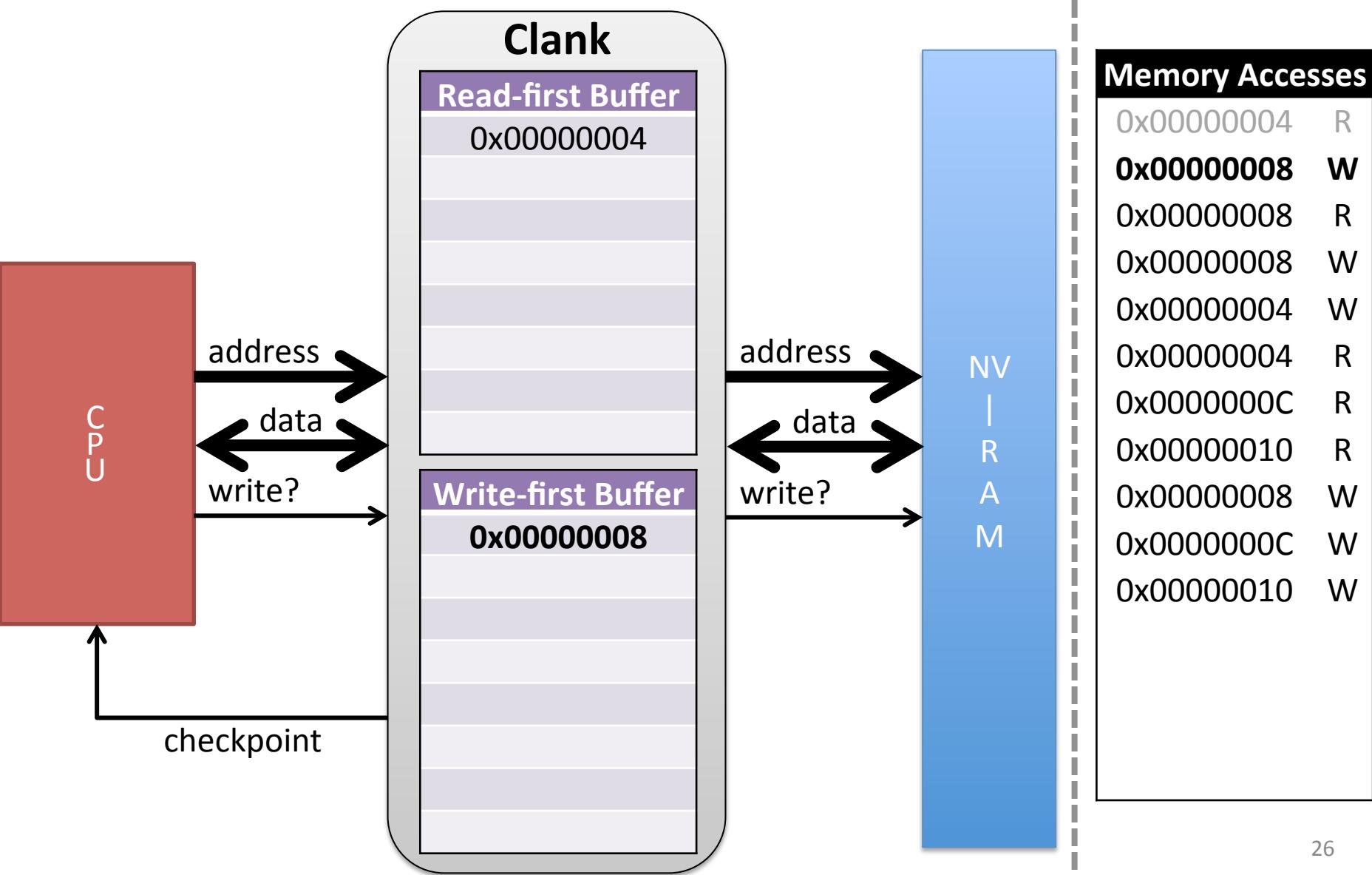
Capacity limited

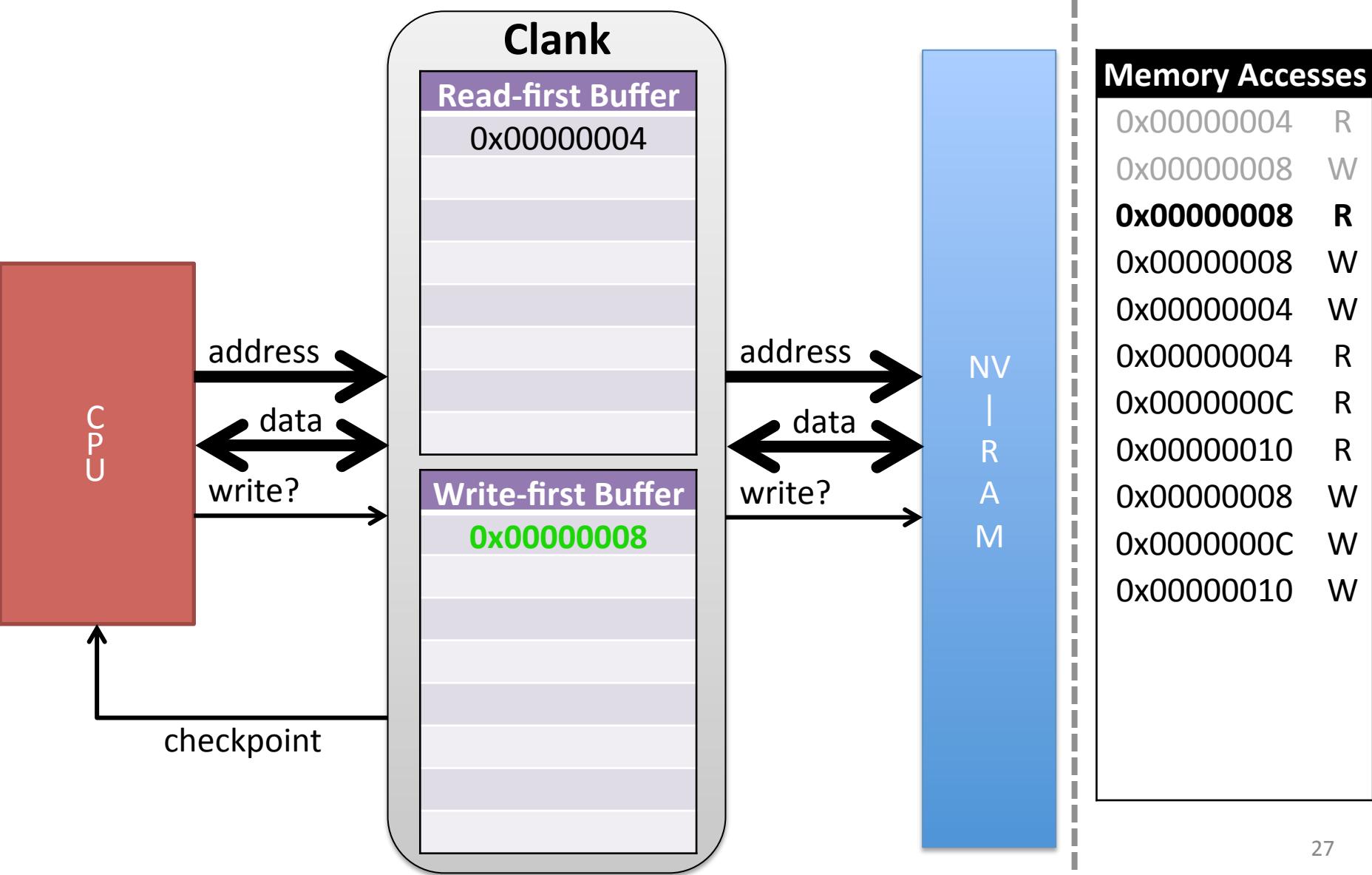


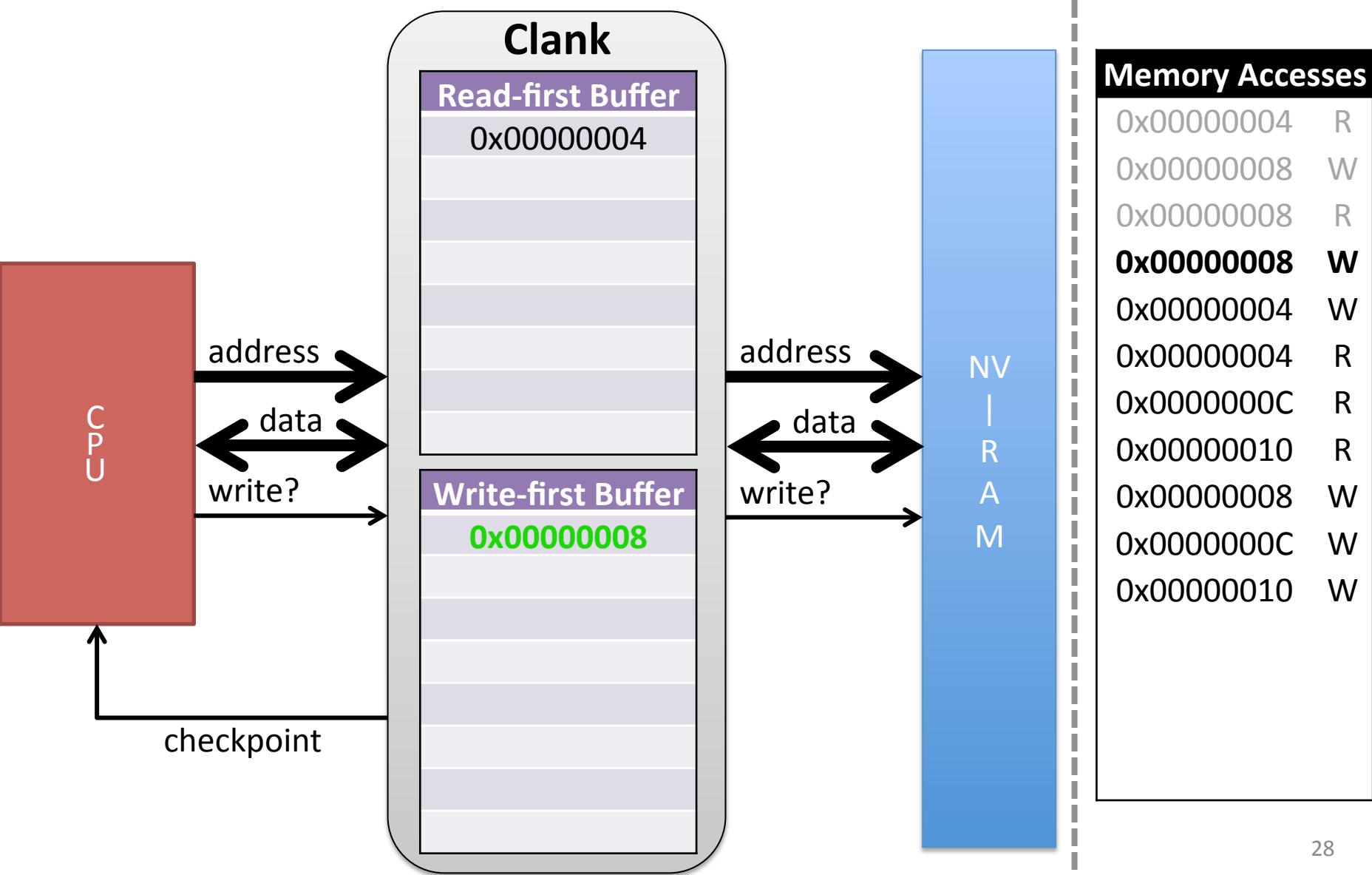
False-violation limited

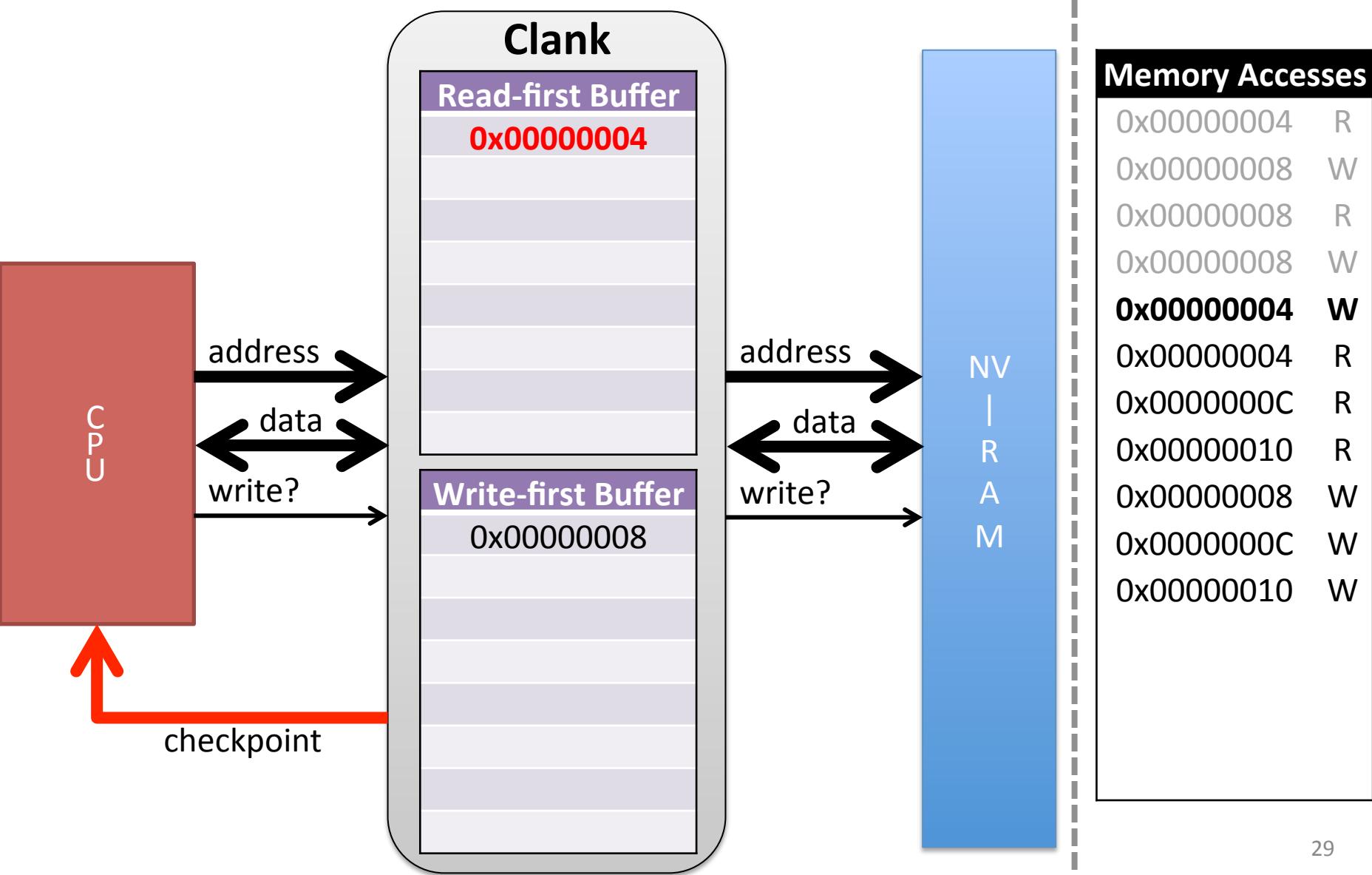


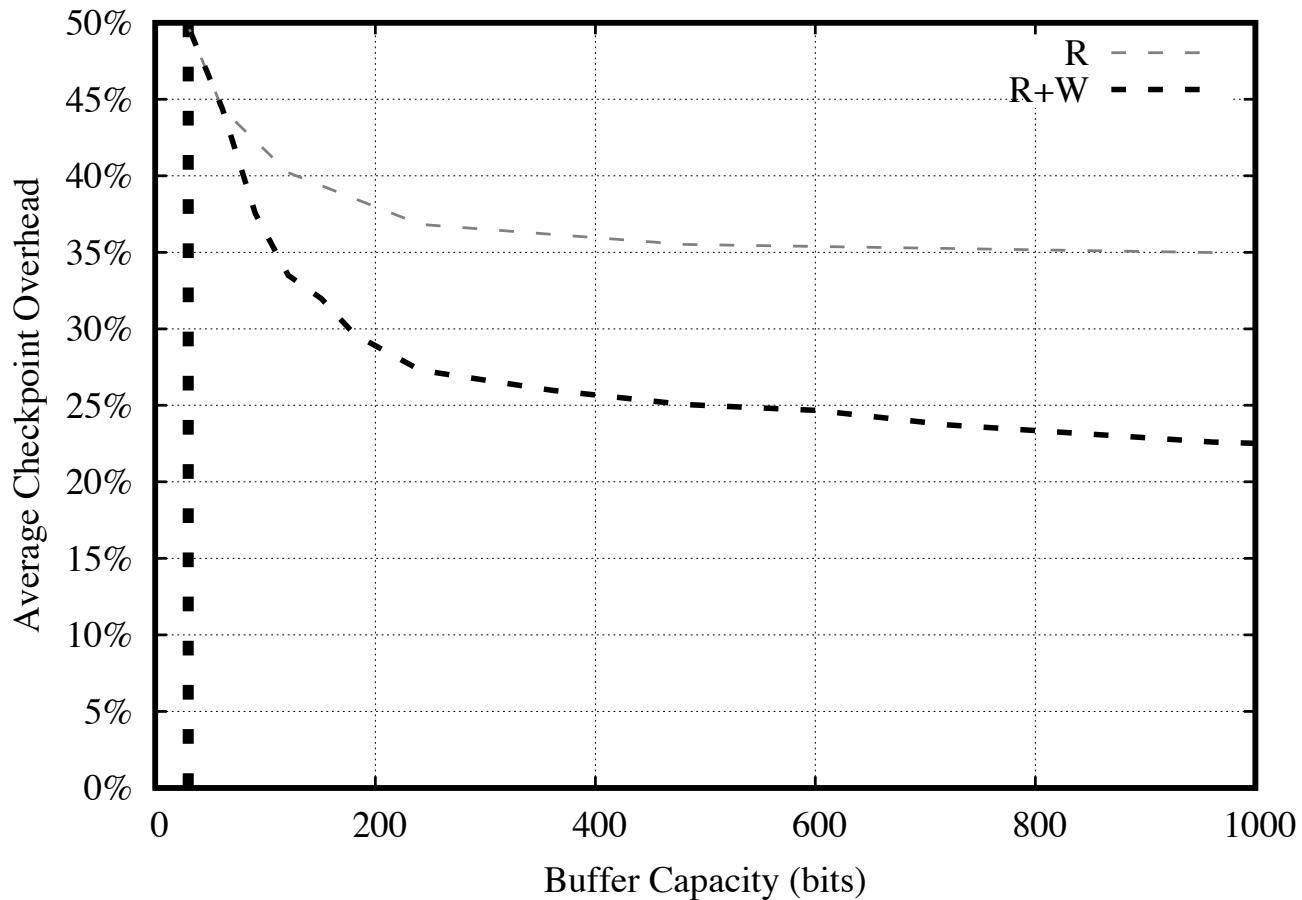




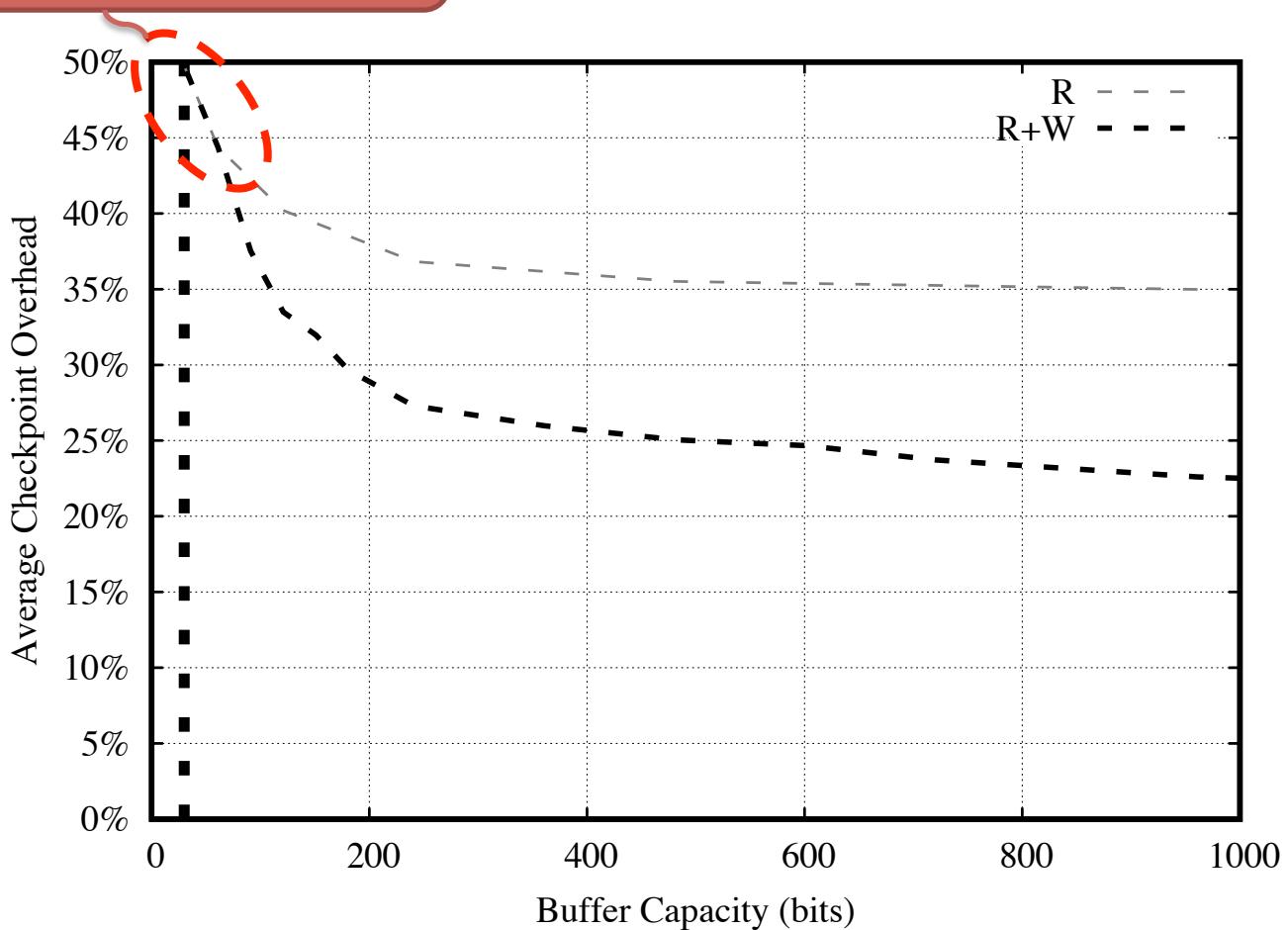




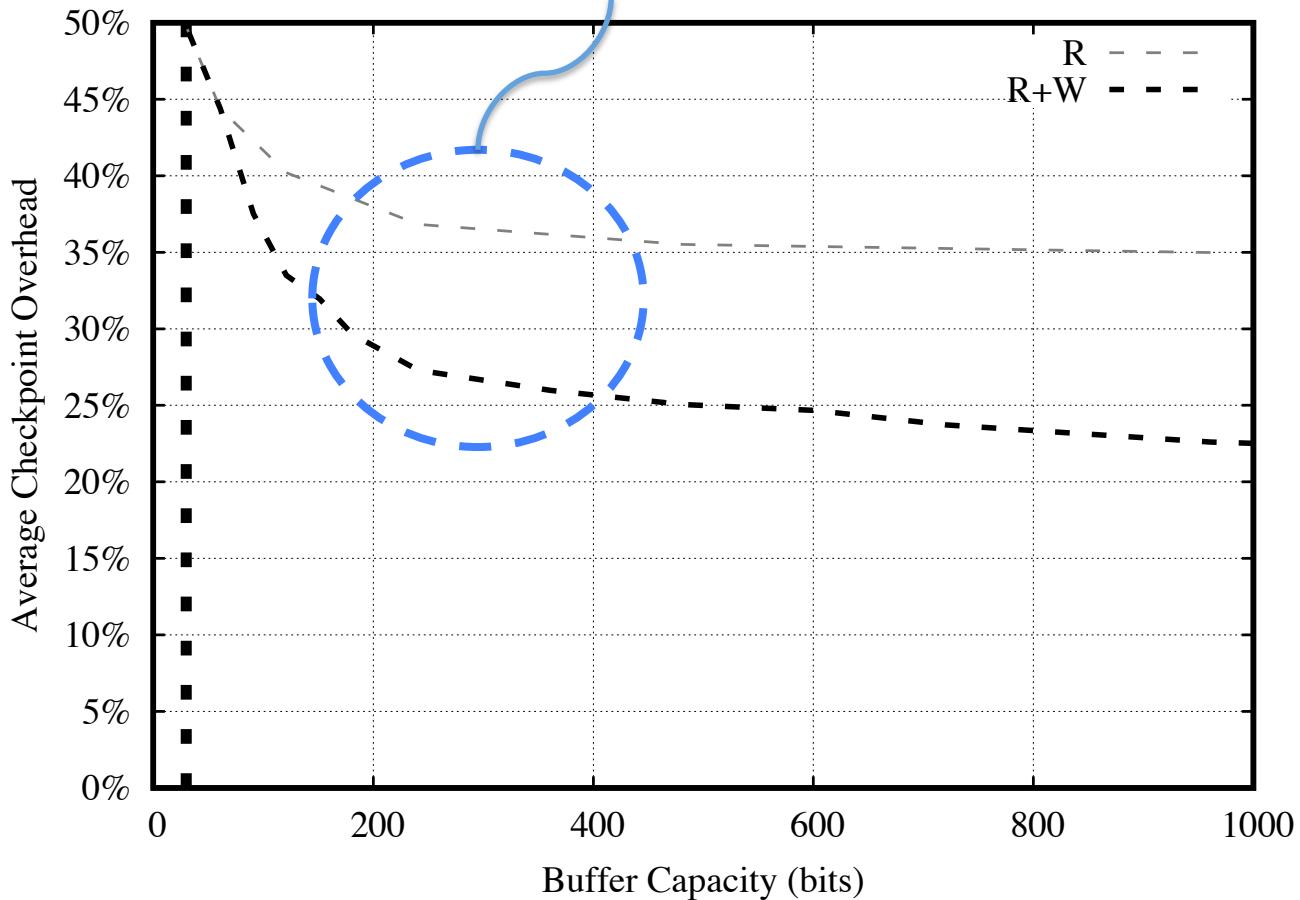




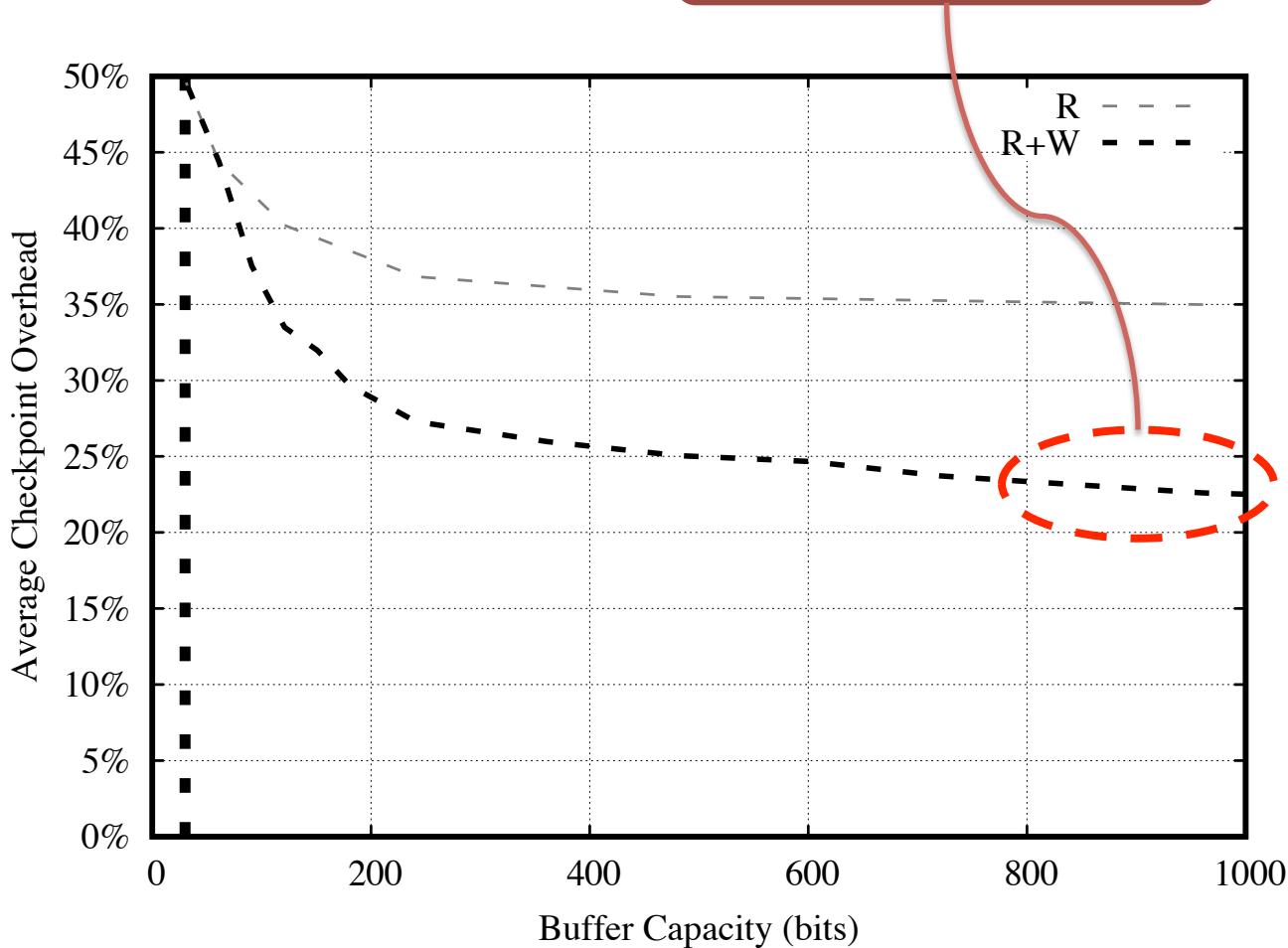
Best to have at two RFB entries
before adding a WFB entry



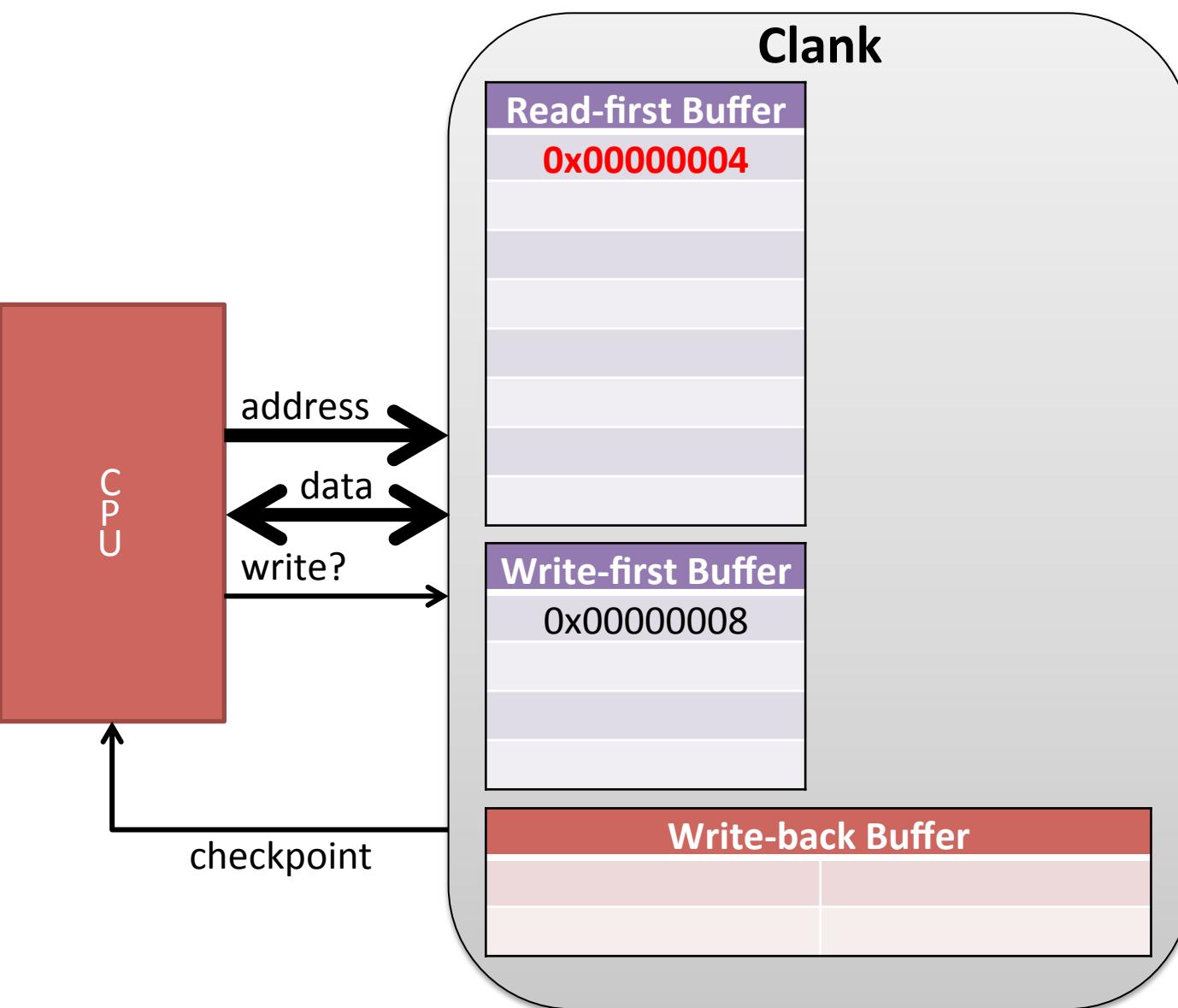
26% overhead reduction
for small configurations



True-violation limited



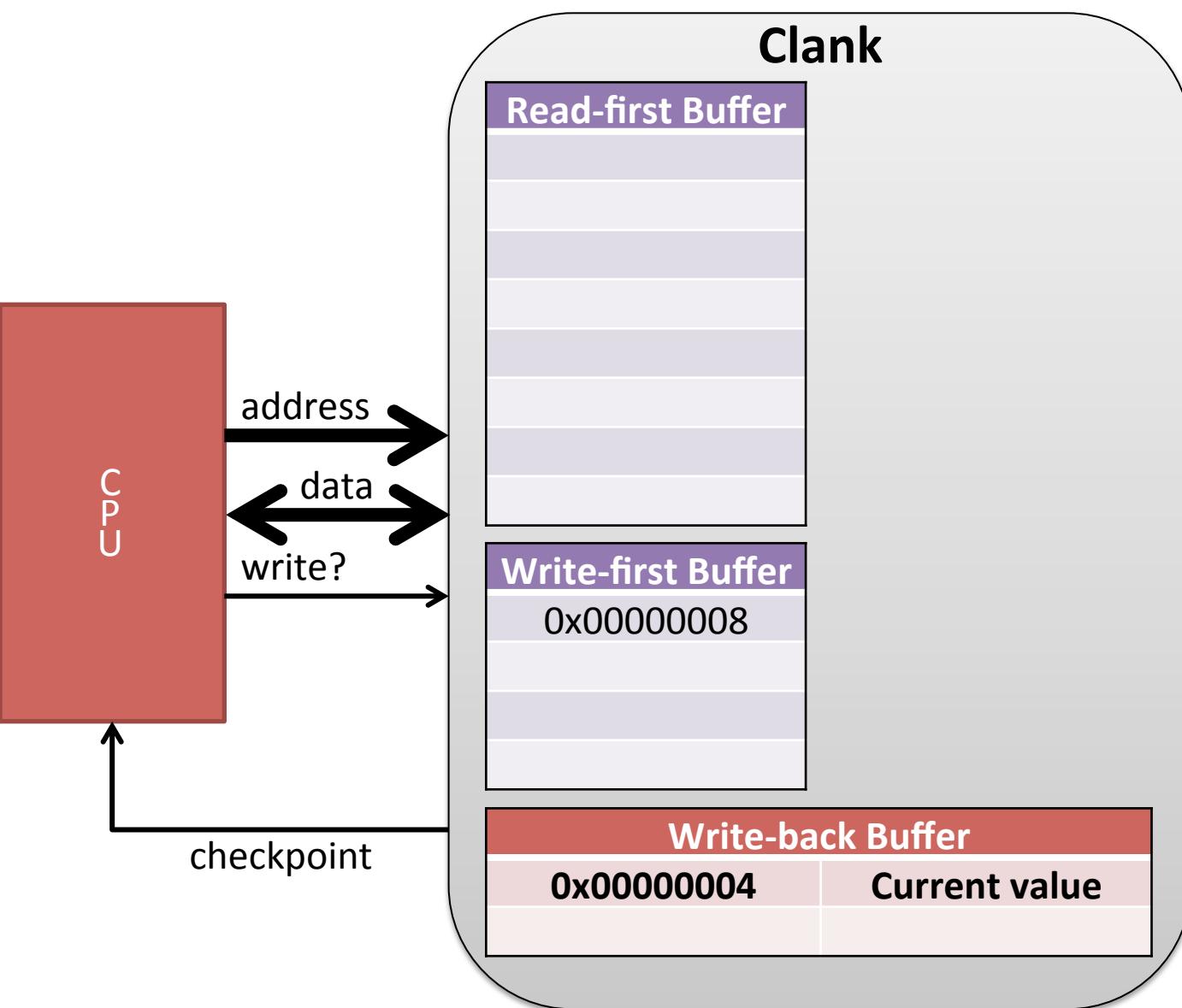
Clank



Memory Accesses

0x00000004	R
0x00000008	W
0x00000008	R
0x00000008	W
0x00000004	W
0x00000004	R
0x0000000C	R
0x00000010	R
0x00000008	W
0x0000000C	W
0x00000010	W

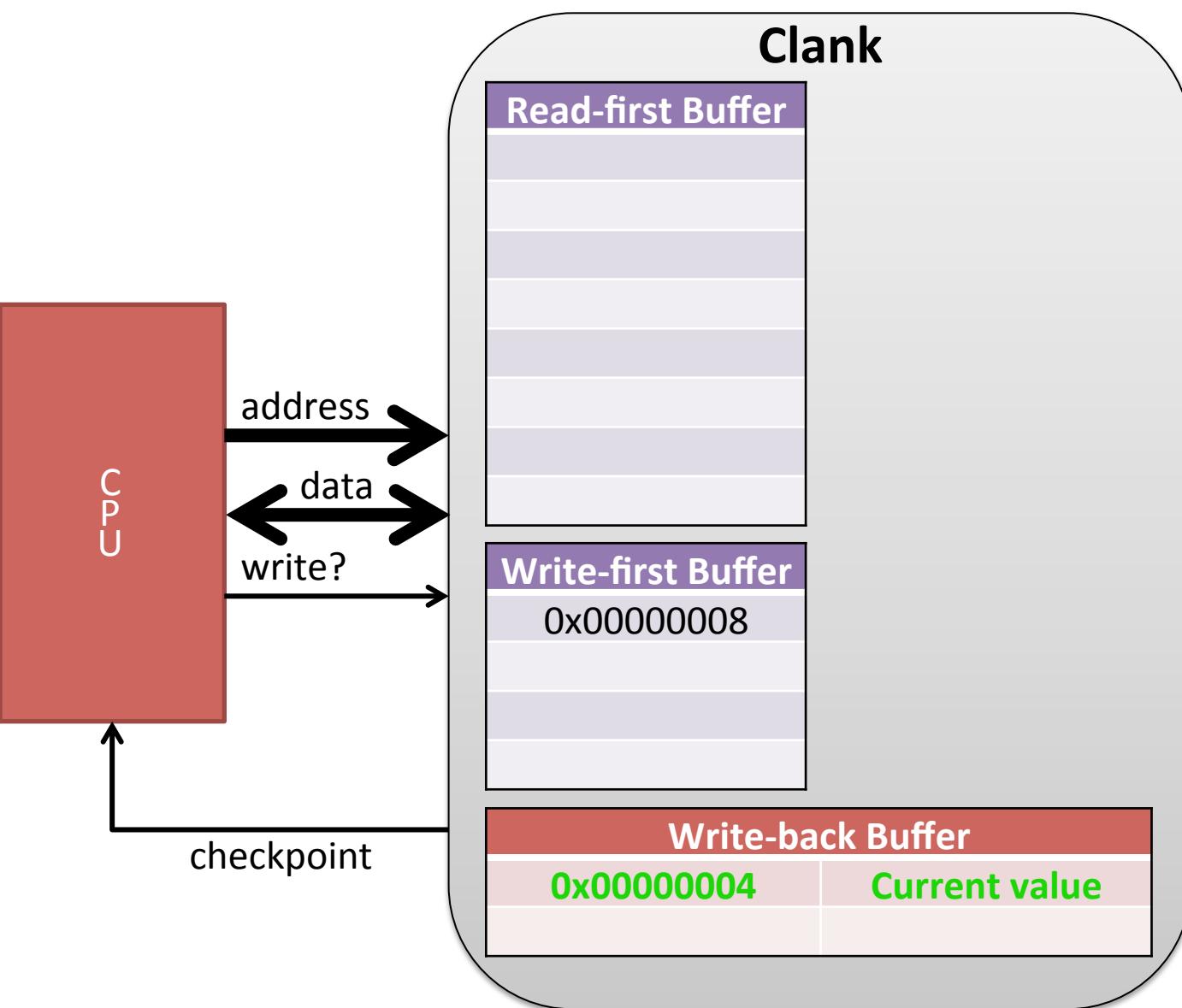
Clank



Memory Accesses

0x00000004	R
0x00000008	W
0x00000008	R
0x00000008	W
0x00000004	W
0x00000004	R
0x0000000C	R
0x00000010	R
0x00000008	W
0x0000000C	W
0x00000010	W

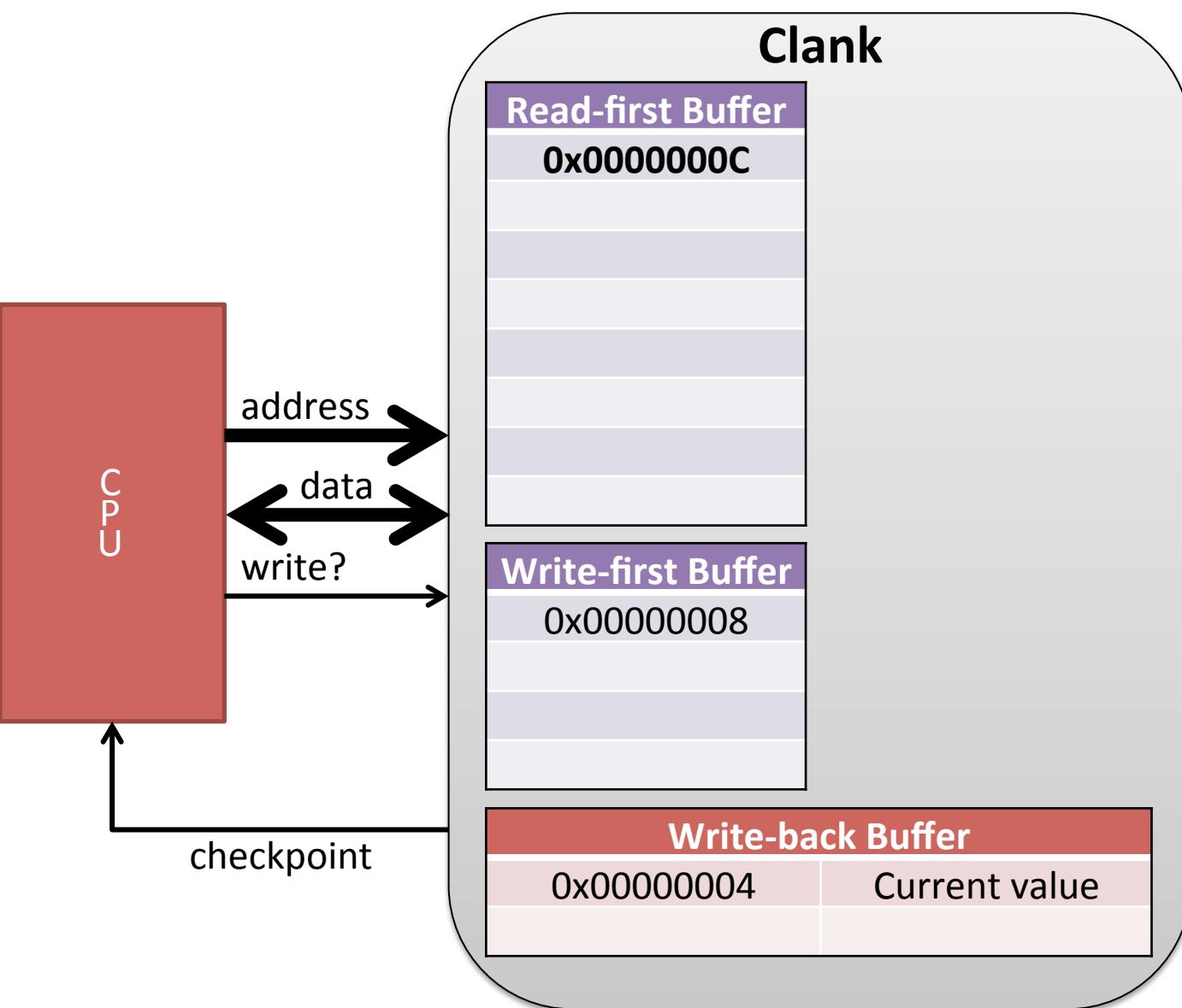
Clank



Memory Accesses

0x00000004	R
0x00000008	W
0x00000008	R
0x00000008	W
0x00000004	W
0x00000004	R
0x0000000C	R
0x00000010	R
0x00000008	W
0x0000000C	W
0x00000010	W

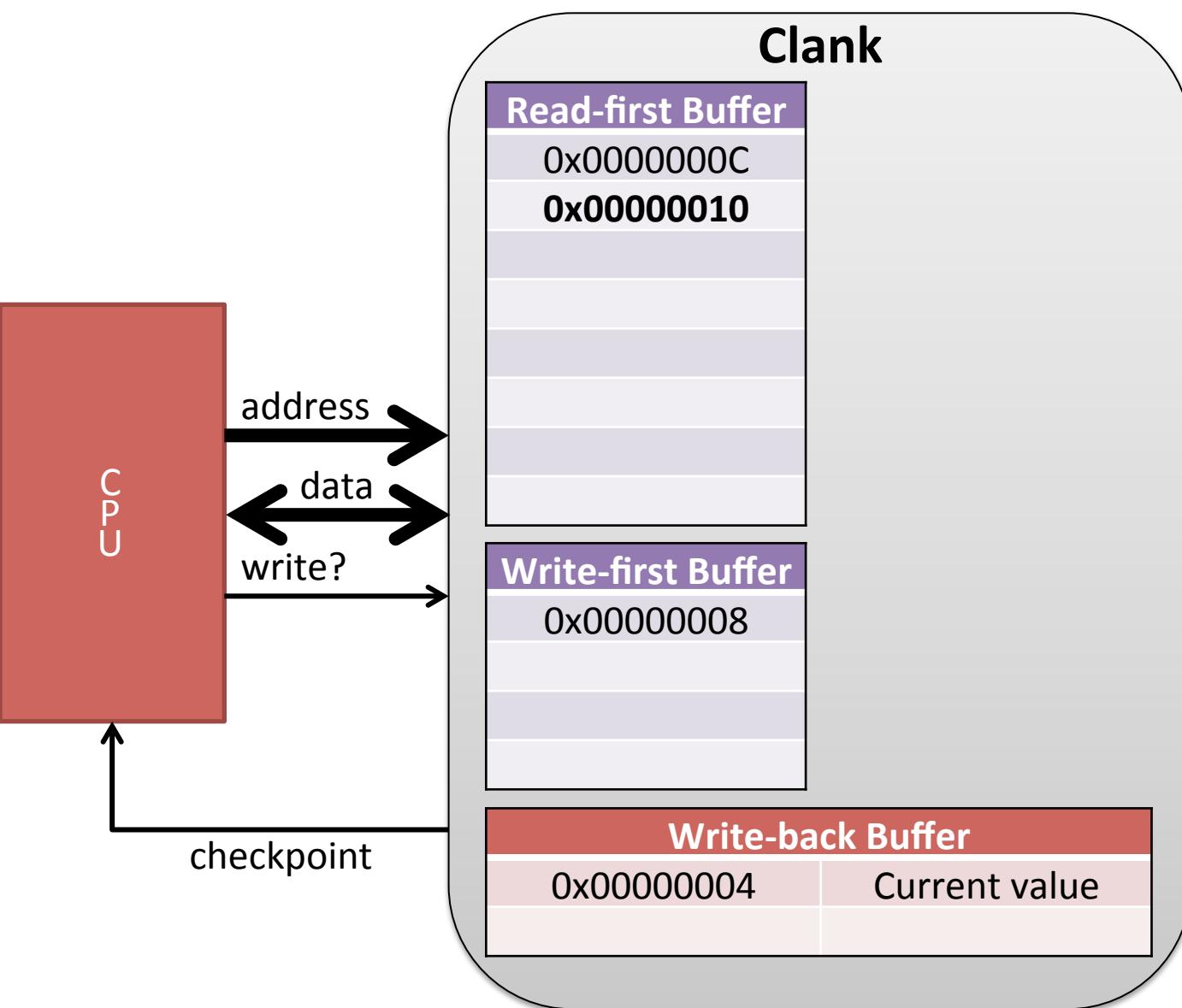
Clank



Memory Accesses

0x00000004	R
0x00000008	W
0x00000008	R
0x00000008	W
0x00000004	W
0x00000004	R
0x0000000C	R
0x00000010	R
0x00000008	W
0x0000000C	W
0x00000010	W

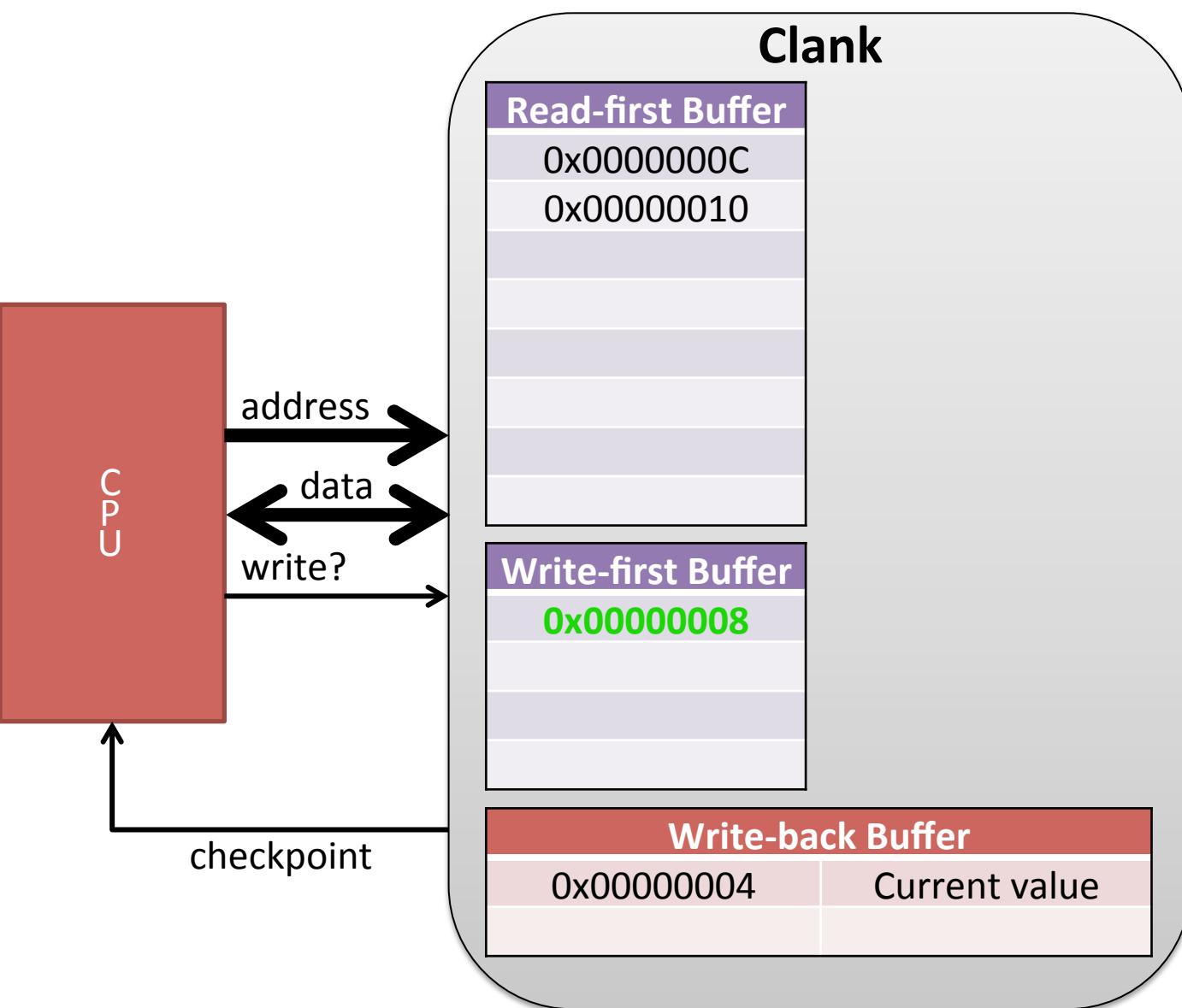
Clank



Memory Accesses

0x00000004	R
0x00000008	W
0x00000008	R
0x00000008	W
0x00000004	W
0x00000004	R
0x0000000C	R
0x00000010	R
0x00000008	W
0x0000000C	W
0x00000010	W

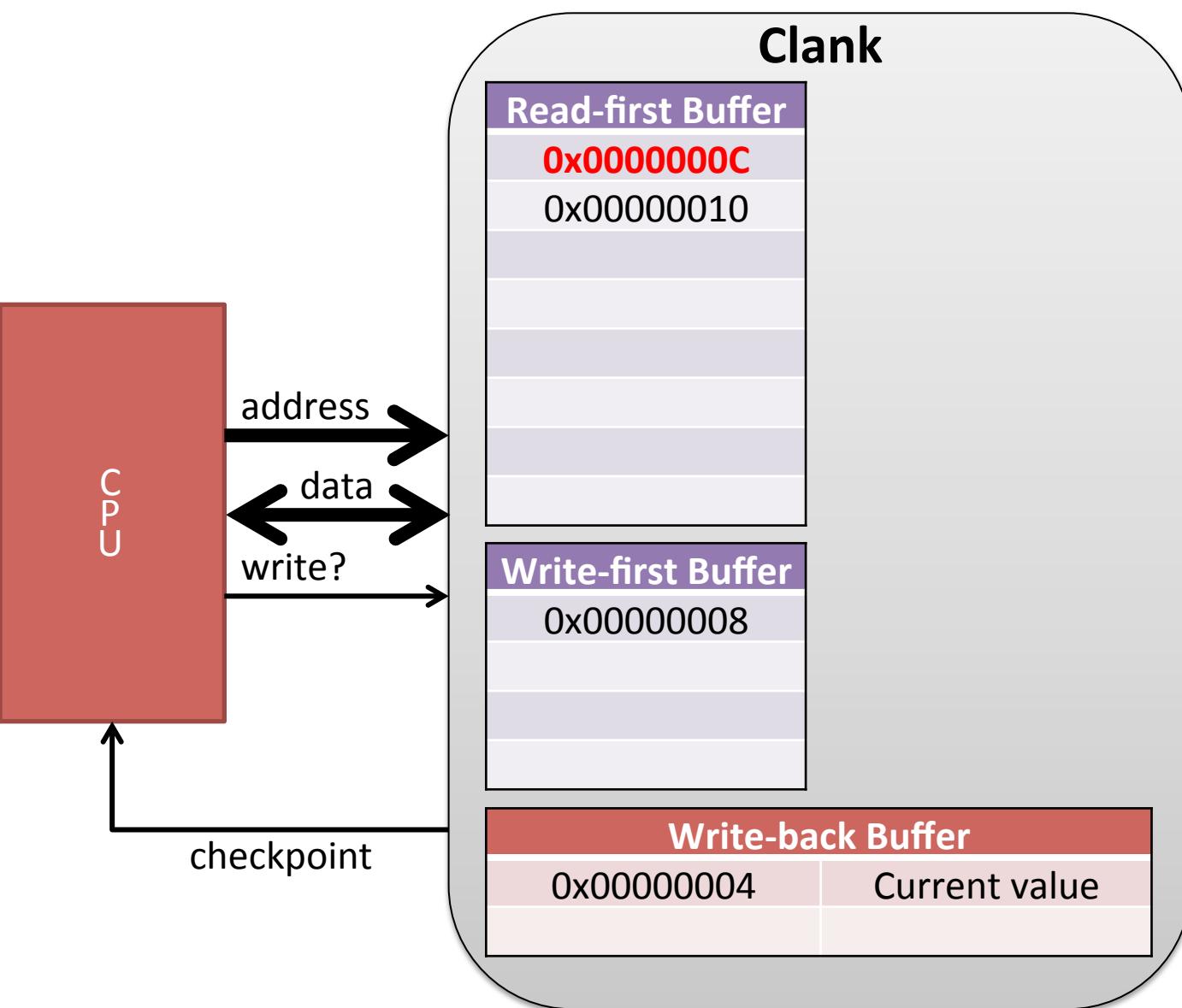
Clank



Memory Accesses

0x00000004	R
0x00000008	W
0x00000008	R
0x00000008	W
0x00000004	W
0x00000004	R
0x0000000C	R
0x00000010	R
0x00000008	W
0x0000000C	W
0x00000010	W

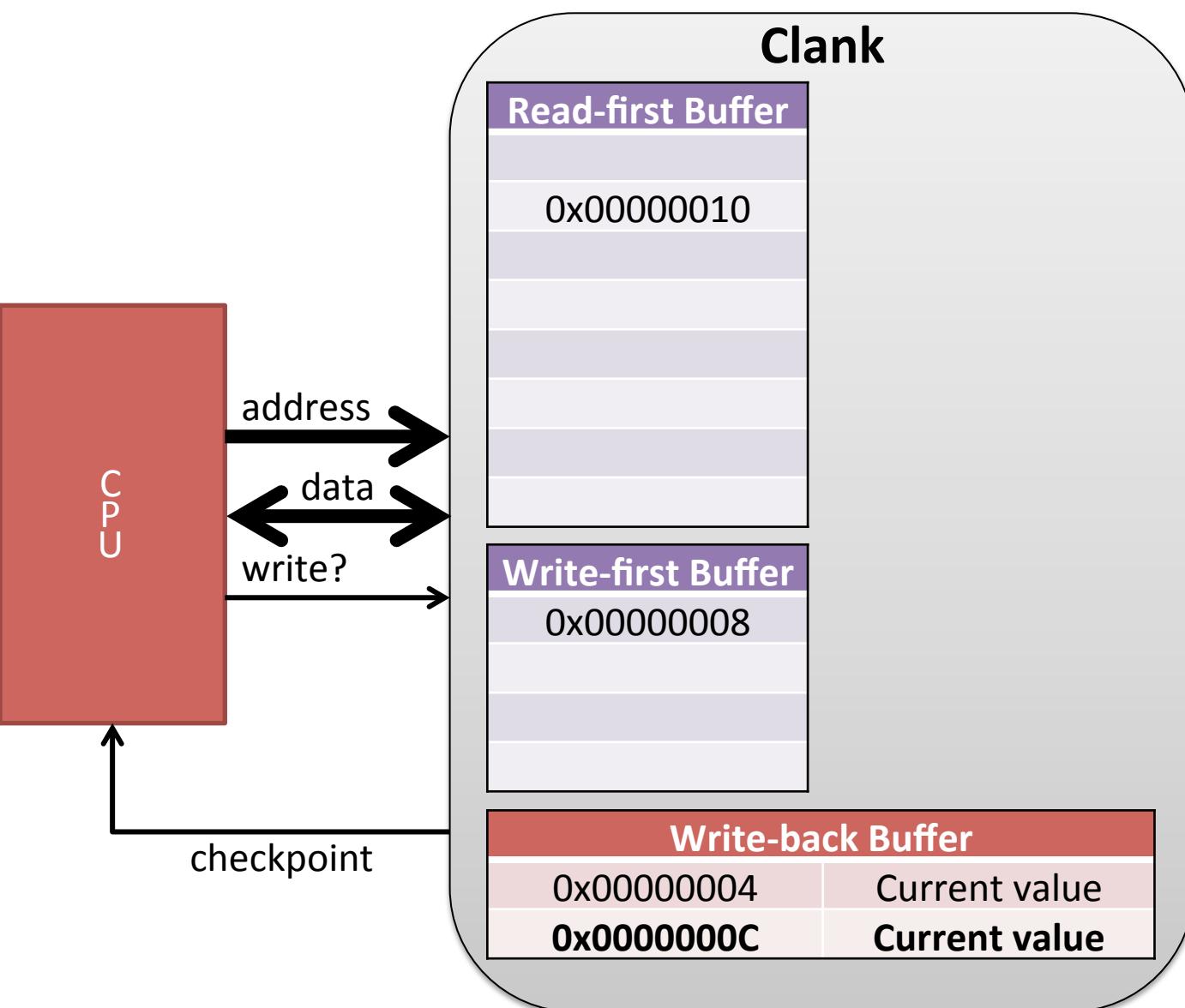
Clank



Memory Accesses

0x00000004	R
0x00000008	W
0x00000008	R
0x00000008	W
0x00000004	W
0x00000004	R
0x0000000C	R
0x00000010	R
0x00000008	W
0x0000000C	W
0x00000010	W

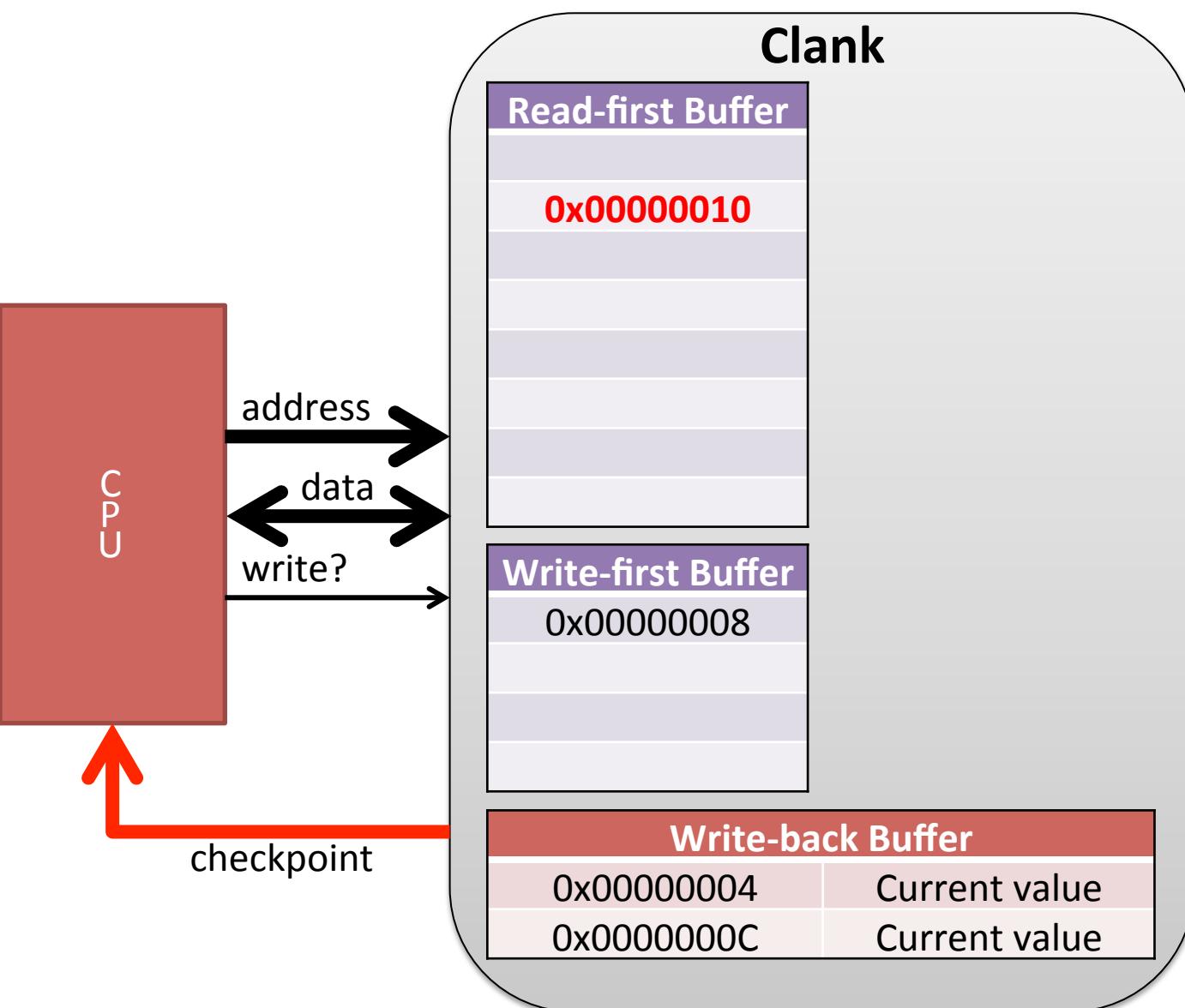
Clank



Memory Accesses

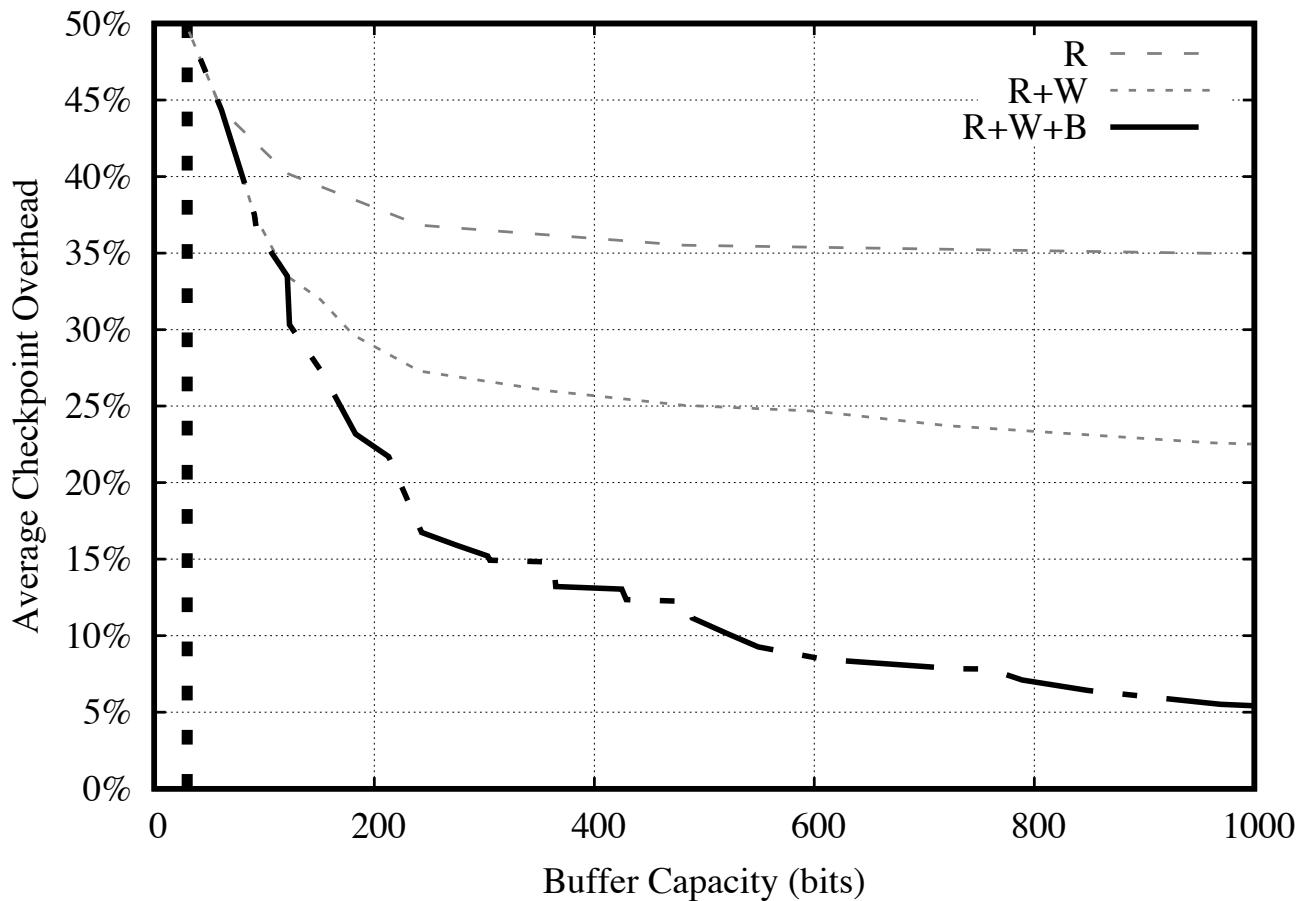
0x00000004	R
0x00000008	W
0x00000008	R
0x00000008	W
0x00000004	W
0x00000004	R
0x0000000C	R
0x00000010	R
0x00000008	W
0x0000000C	W
0x00000010	W

Clank

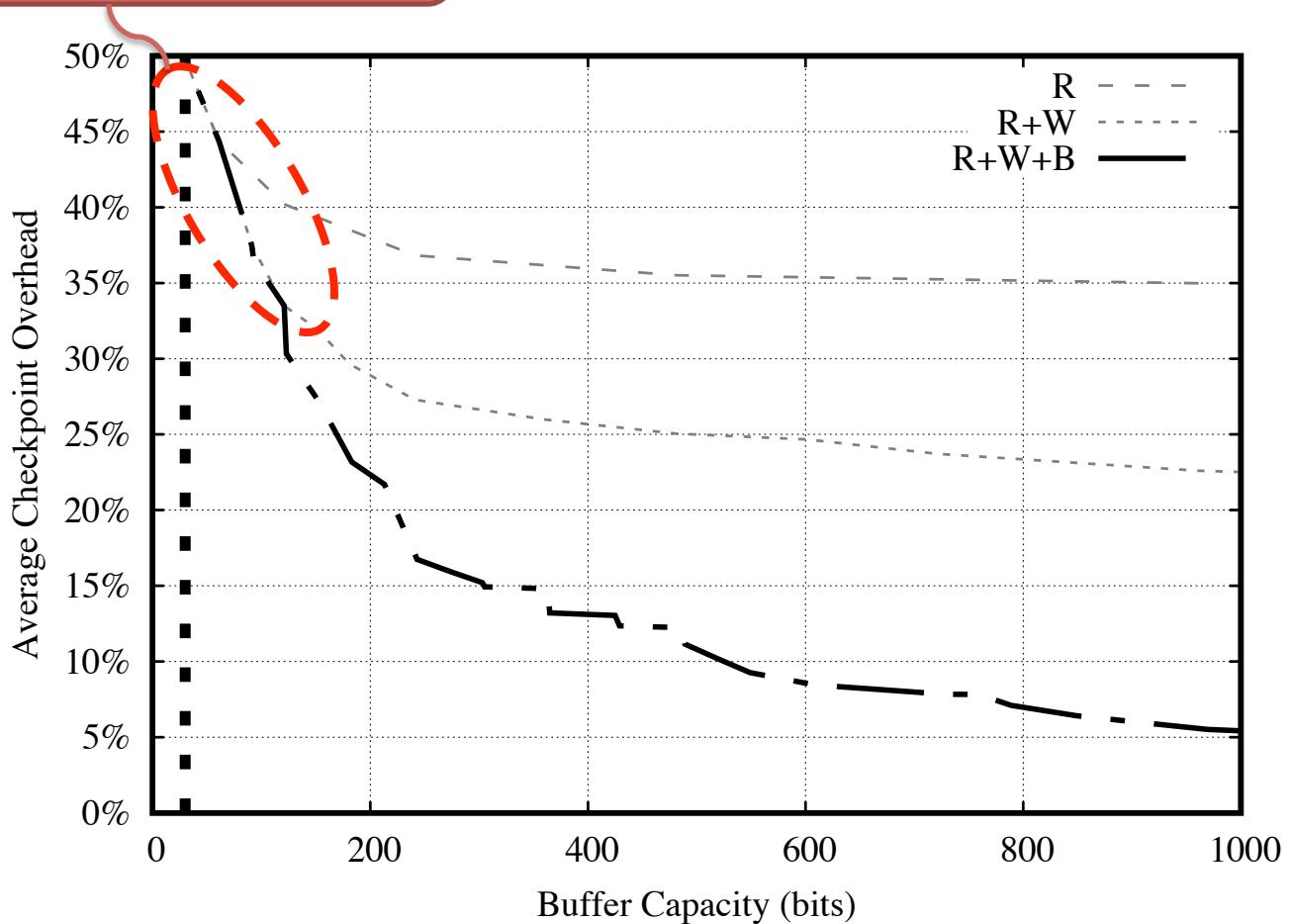


Memory Accesses

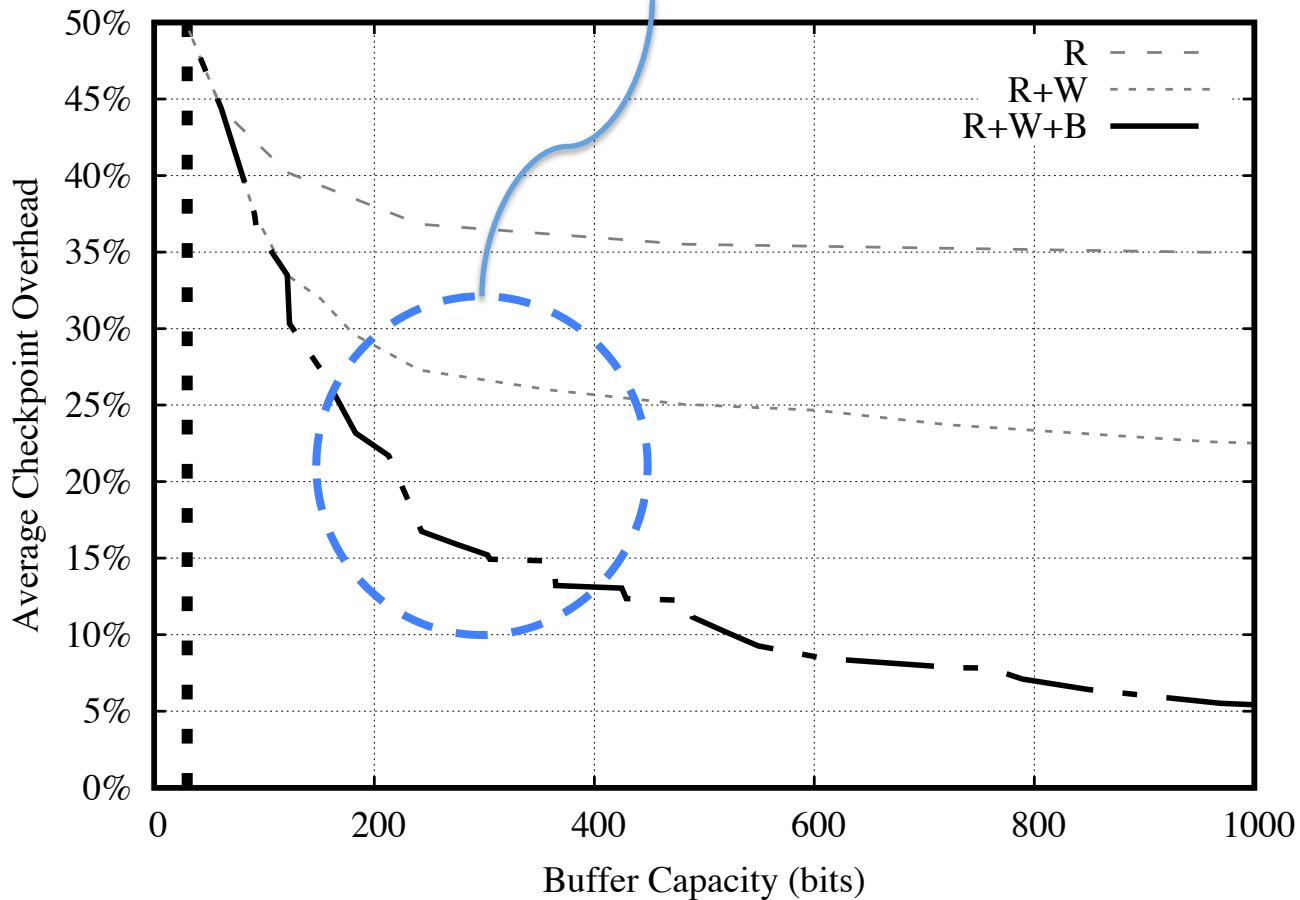
0x00000004	R
0x00000008	W
0x00000008	R
0x00000008	W
0x00000004	W
0x00000004	R
0x0000000C	R
0x00000010	R
0x00000008	W
0x0000000C	W
0x00000010	W



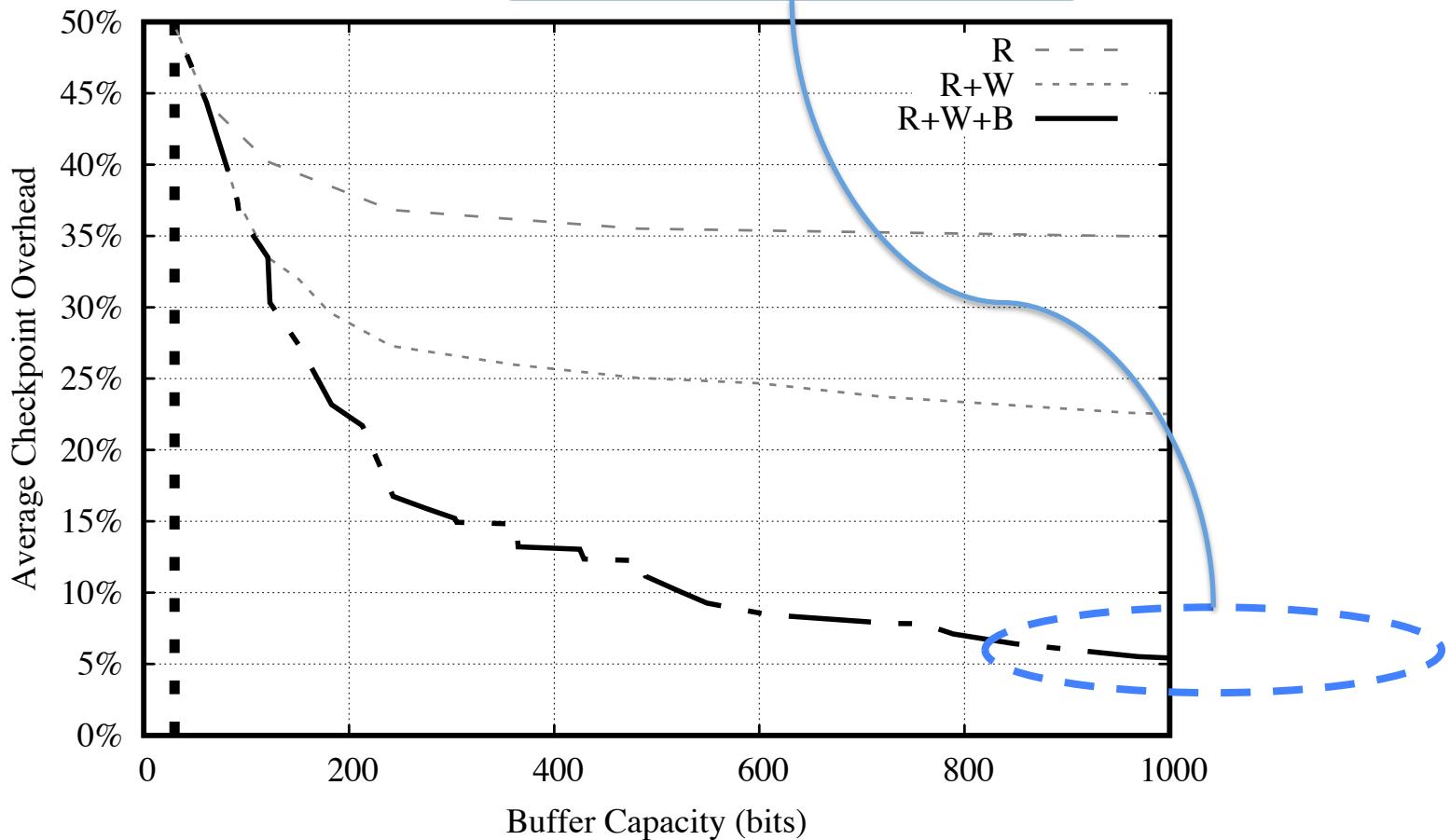
Only add WBB entries when RFB
and WFB have excess capacity



Additional 20% to 48%
overhead reduction for small
configurations



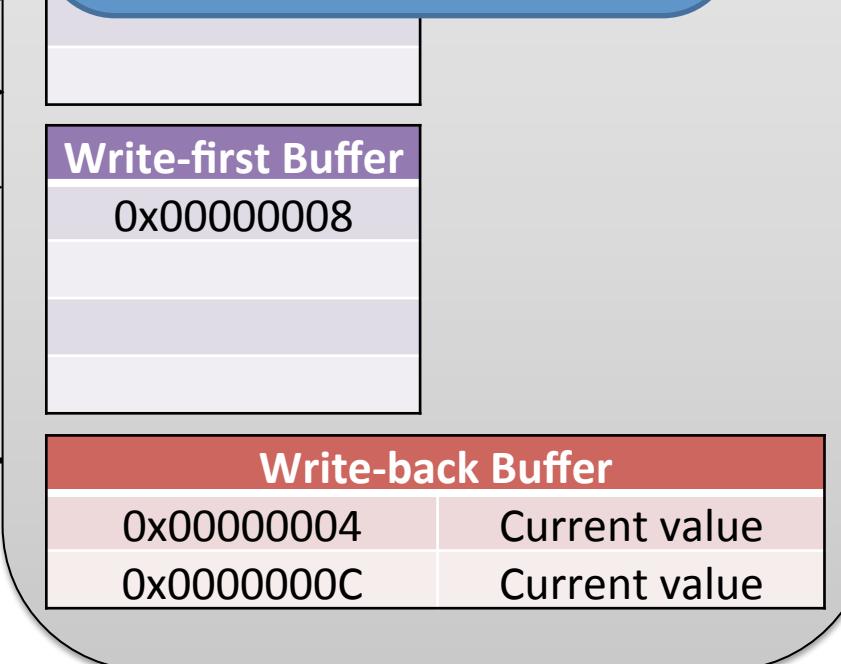
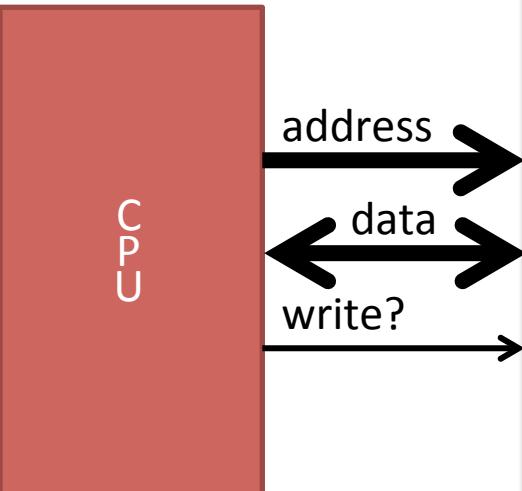
Approaches 0% overhead
...eventually



Clank

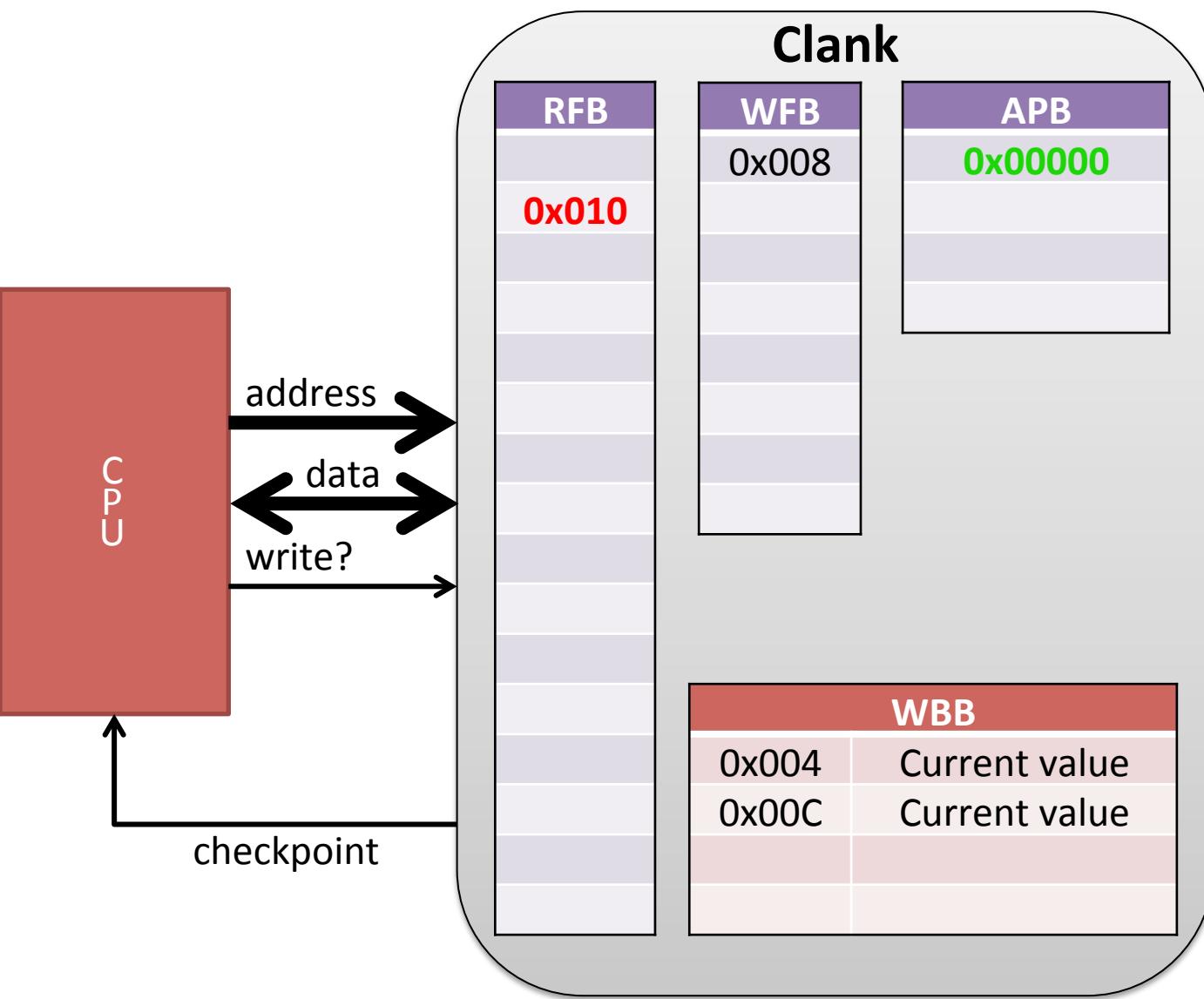
Memory accesses have high spatial locality

Let's de-duplicate!



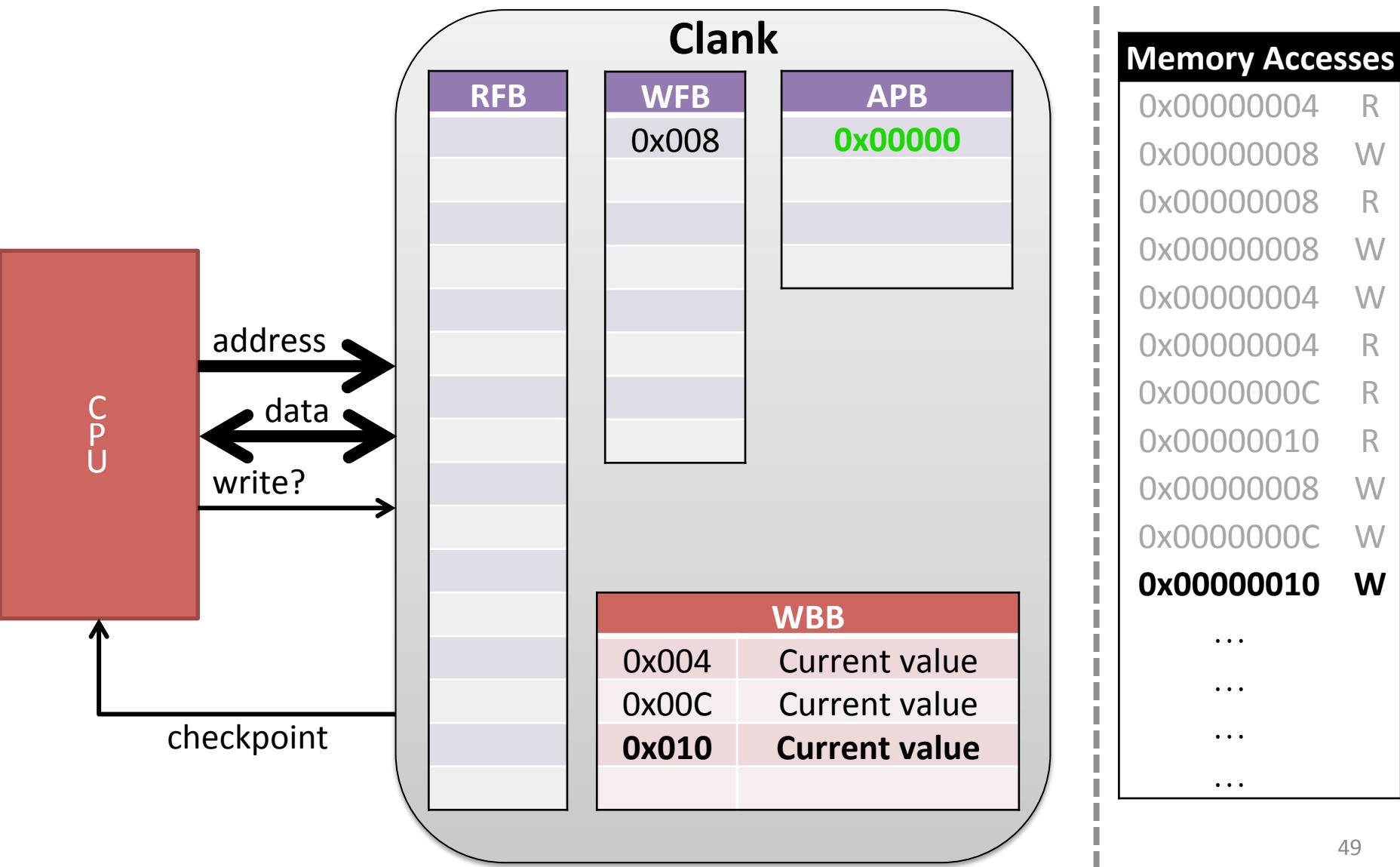
Memory Accesses

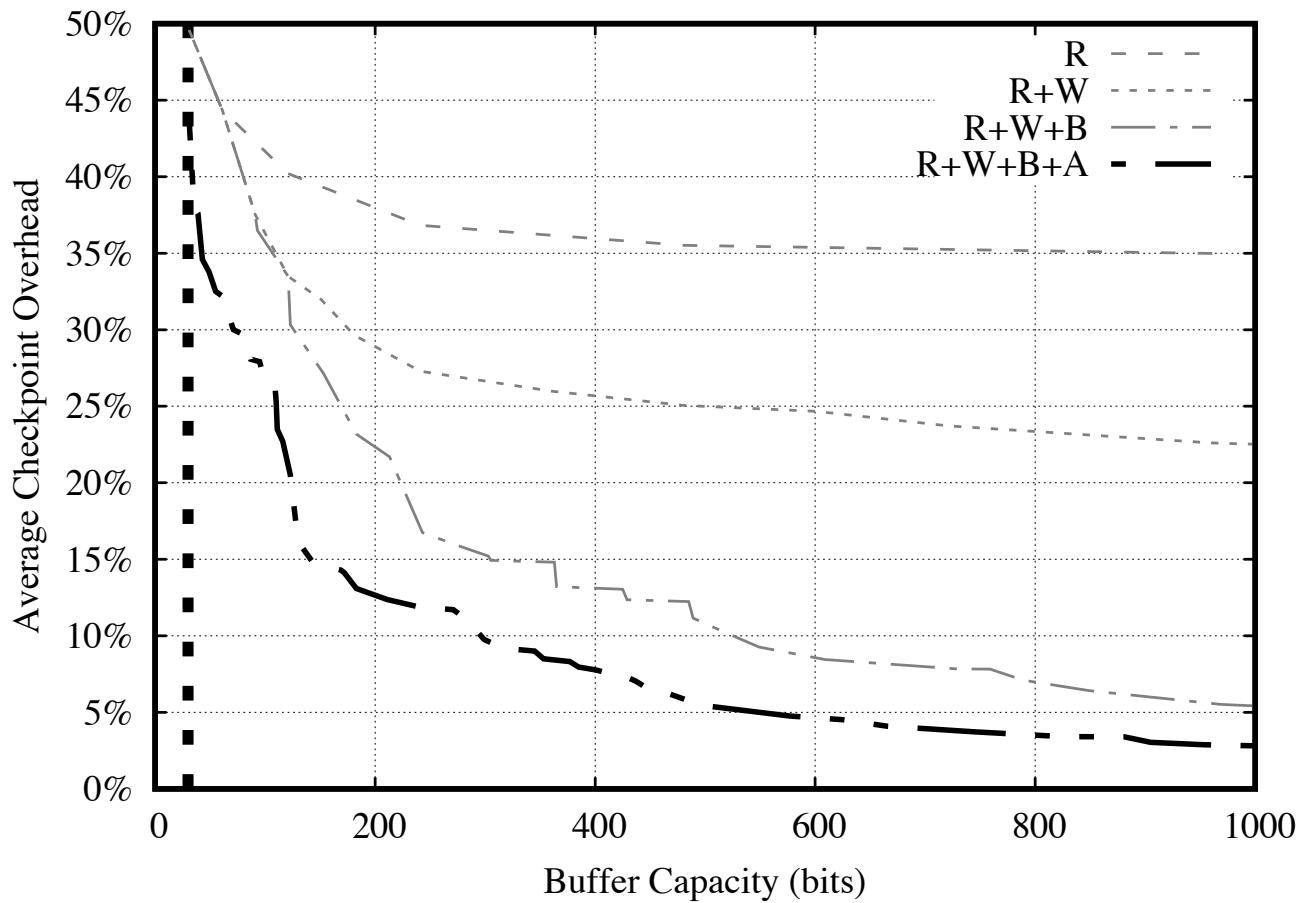
0x00000004	R
0x00000008	W
0x00000008	R
0x00000008	W
0x00000004	W
0x00000004	R
0x00000004	R
0x0000000C	R
0x00000010	R
0x00000008	W
0x0000000C	W
0x00000010	W



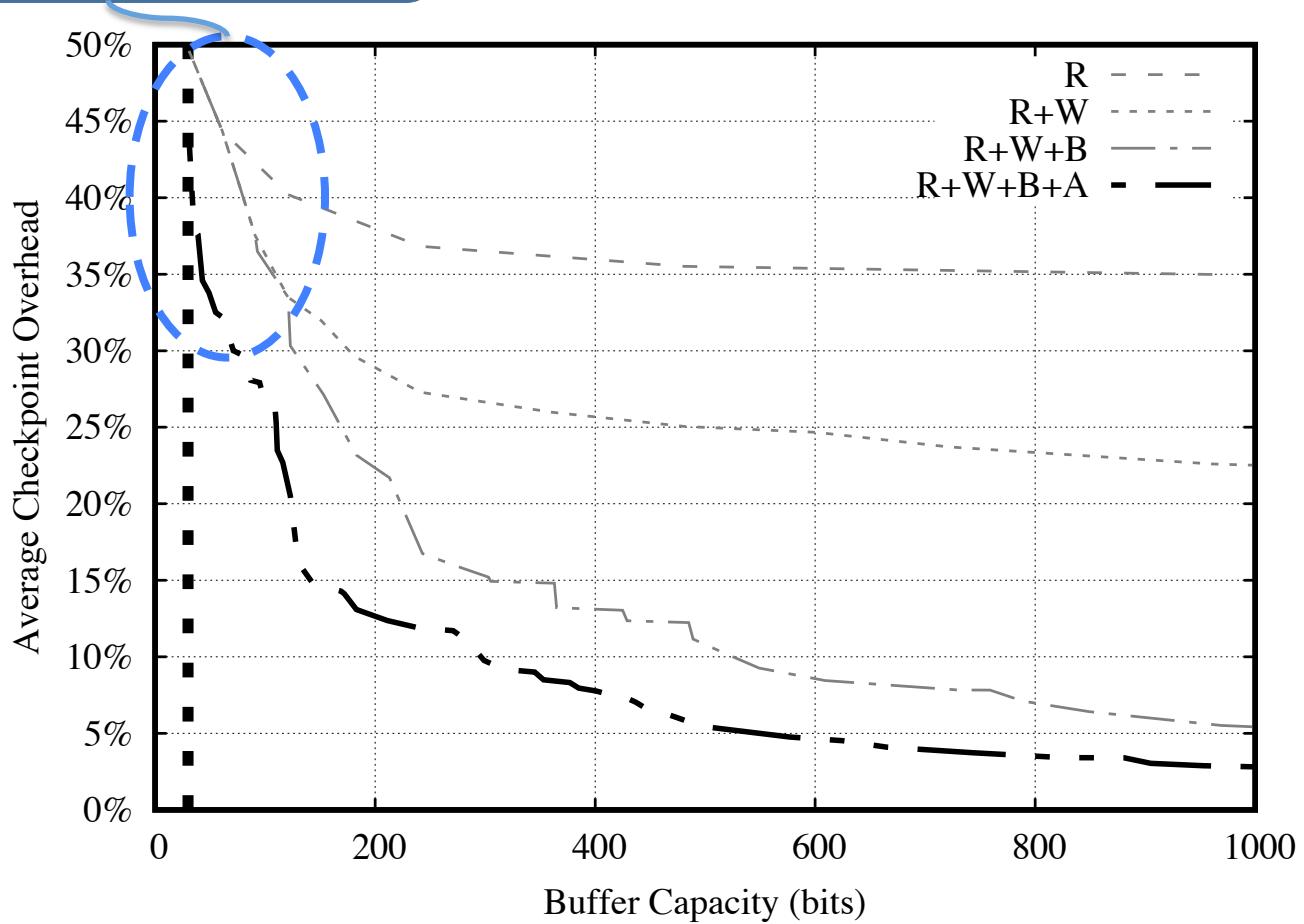
Memory Accesses

0x00000004	R
0x00000008	W
0x00000008	R
0x00000008	W
0x00000004	W
0x00000004	R
0x0000000C	R
0x00000010	R
0x00000008	W
0x0000000C	W
0x00000010	W
...	
...	
...	
...	

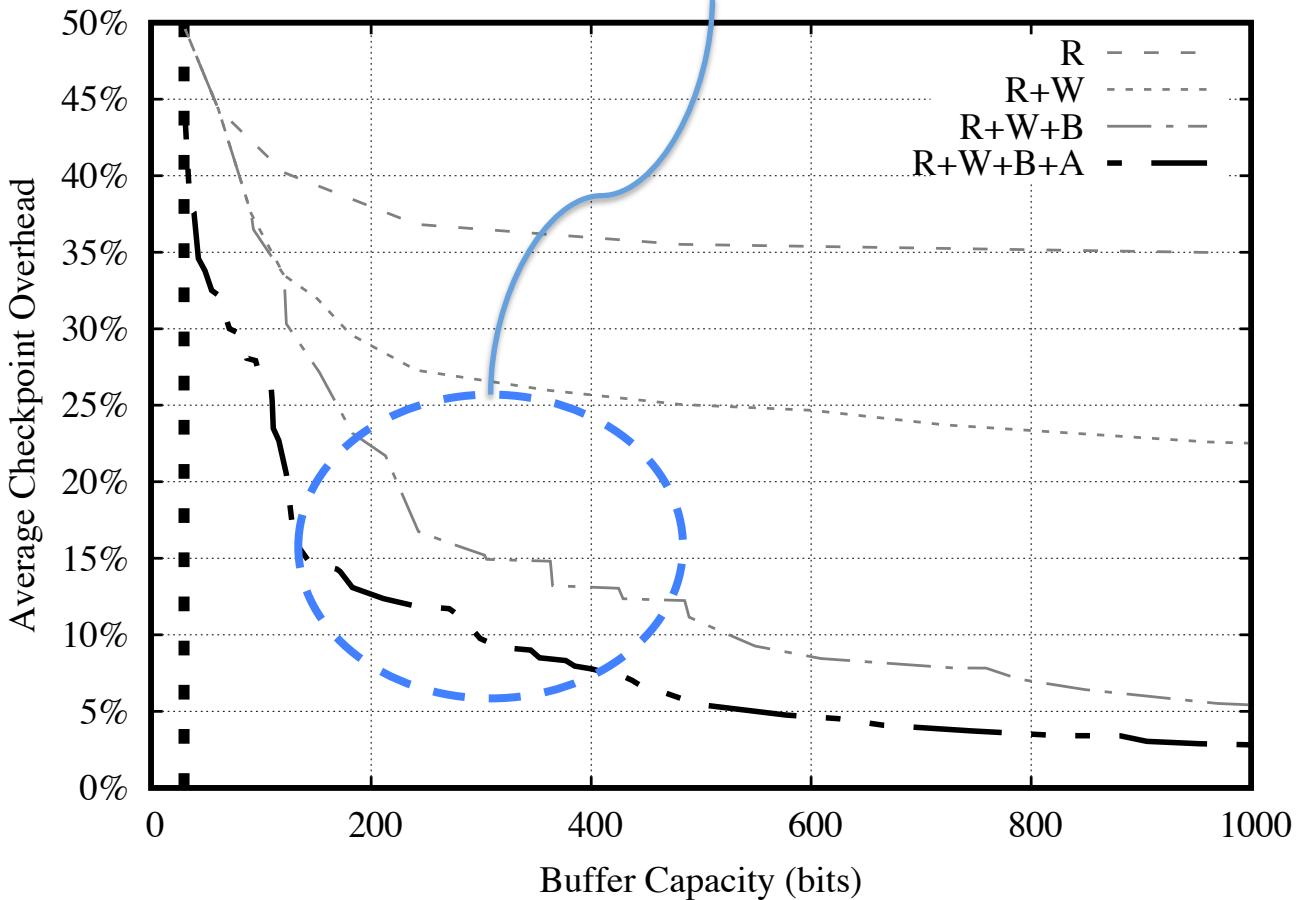




Fundamental change!
Same cost, more entries



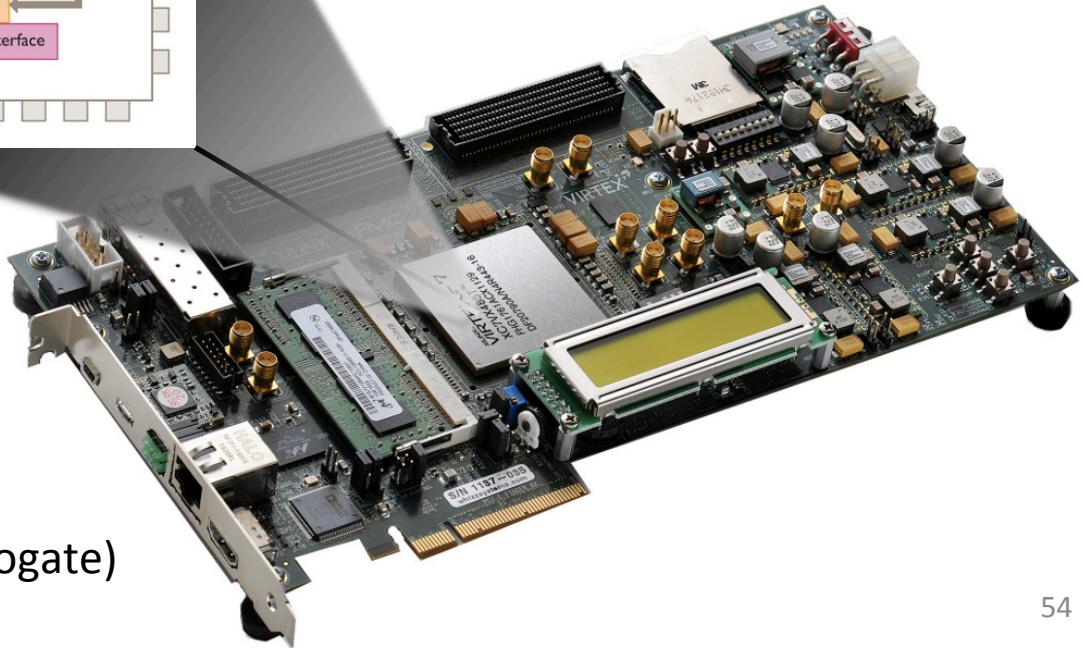
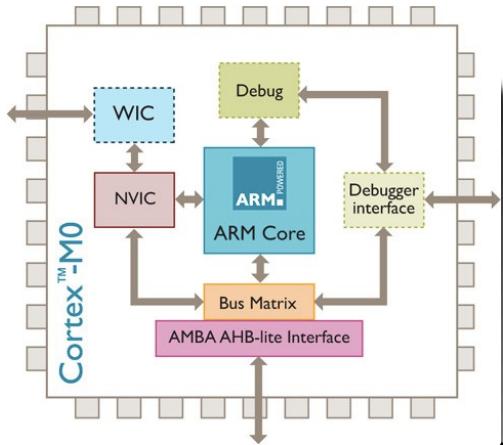
Additional 35% to 43% overhead reduction
for small configurations
---up to 78% total reduction---



More in the paper

- 5 buffer management optimizations
- Performance watchdog timer
- Progress watchdog timer
- Formal verification
- Combining Clank with an aware compiler
- Alternative memory compositions

ARM-based Clank implementation



Targets:

- Verilog
- EBMC Verification
- ARM Cortex-M0
- VC709 FPGA
- 50 MHz
- 128 KB BRAM (as FRAM surrogate)

Clank is lightweight

R, W, WB, AP	LUT	FF	Memory	Avg. HW	Avg. SW	Est. Total
16, 0, 0, 0	2.5%	0.7%	.2%	1.1%	33.8%	34.9%
8, 8, 0, 0	2.4%	0.7%	.2%	1.1%	27.3%	28.4%
8, 4, 2, 0	2.1%	0.7%	.2%	1.0%	15.7%	16.7%
16, 8, 4, 4	3.4%	1.5%	.3%	1.7%	8.0%	9.7%
16, 8, 4, 4 (+C+WDT)	3.4%	1.5%	.3%	1.7%	6.0%	7.7%

Minimum Clank (total) overhead is ~6%.

Clank has the lowest total overhead

FFT at 100ms avg. on time

Approach	Total Overhead	Burden
Dino	Not ported	Programmer
Mementos (on FRAM)	117% - 145%	Voltage measurement*
Hibernus	38%	Voltage measurement
Hibernus++	36%	Voltage measurement
Ratchet	32%	Compiler
Clank	6%	Architecture

More experiments in the paper

- Memory overhead
- Impact of buffer management optimizations
- Per benchmark total run time overhead
- Tradeoff between checkpoint and re-execution overhead
- Clank's performance on Flash-like memory compositions

Clank allows software to correctly and automatically bridge power cycles



Opportunity for Clank-aware software

