The Synchronization Power of Coalesced Memory Accesses

Phuong H. Ha (Univ. of Tromsø, Norway)
Philippas Tsigas (Chalmers Univ. of Tech., Sweden)
Otto J. Anshus (Univ. of Tromsø, Norway)

Problem

- Memory access mechanisms influence the system synchronization capability.

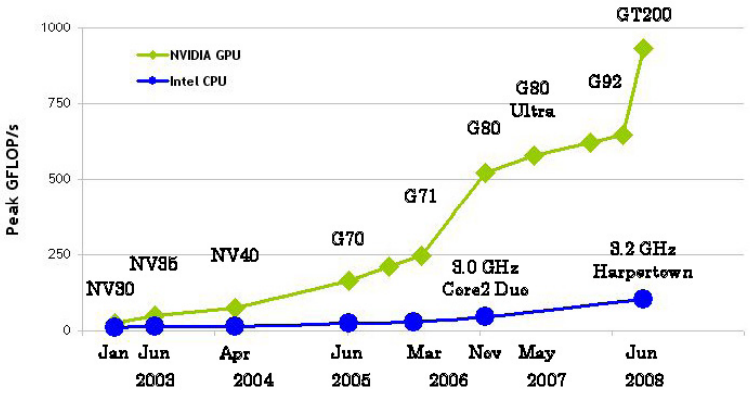
- Conventional wisdom: single-word assignment has consensus number 1

  $\Rightarrow$ stronger synch. primitives (e.g. TAS, FAA, CAS) added.

- Can we make single-word assignment stronger?

  $\Rightarrow$ transistors saved from strong synch. primitives can be used to enhance other functionality.

Transistor distribution



[These figures are from NVIDA CUDA Programming Guide, version 2.0]

What is a memory word?

- A group of *n* bytes that can be stored or retrieved in a single, basic operation.
  - *n* is called *word size*

  (in byte-addressable memory)

- Words of size *n* must always start at addresses that are multiples of *n*.

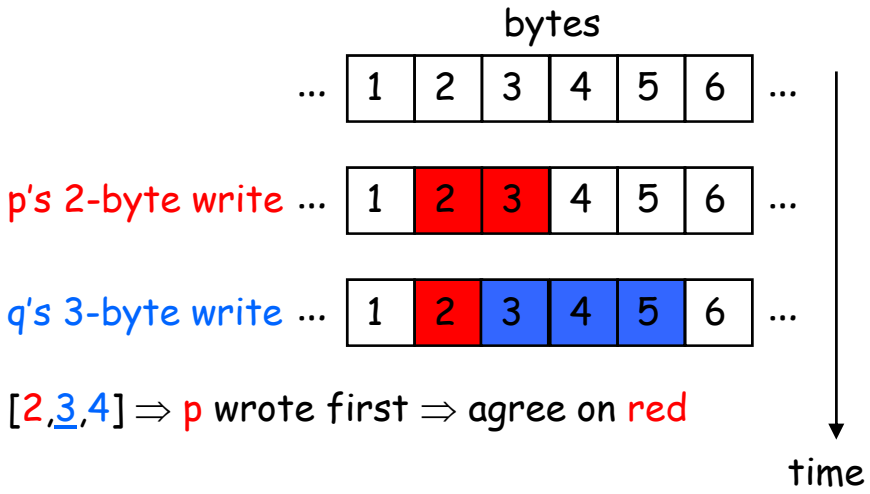
  (Alignment restriction)

[Hamacher et al. 2002, Hennessy et al. 2003]

Key idea 1

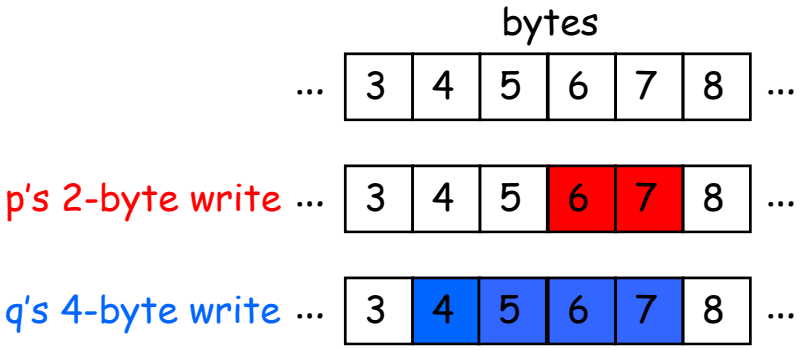
- Word size *n* can be any integer
  - instead of powers of 2 as in conventional architectures
  - Ex: solving 2-process consensus using 2-byte write and 3-byte write.
  - Feasibility: NVIDIA CUDA
    - *int1, int2, int3, int4*

Size-varying word model (svword)



[2,3,4] ⇒ p wrote first ⇒ agree on red

Conventional architectures

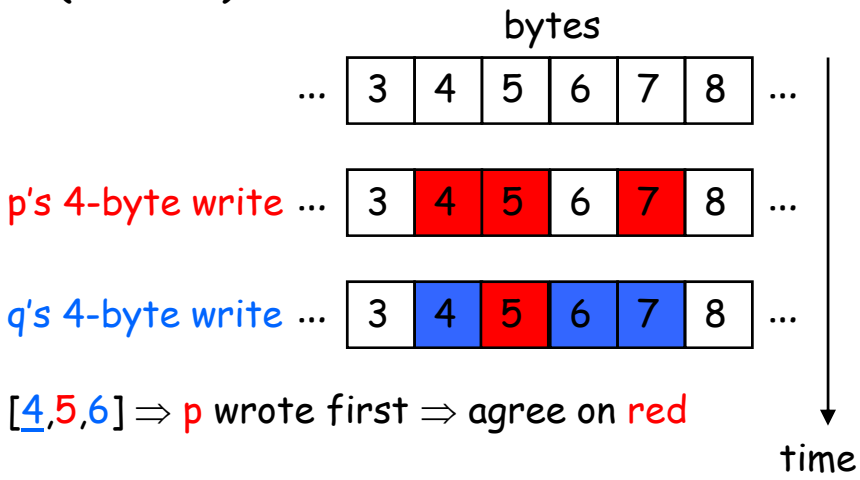


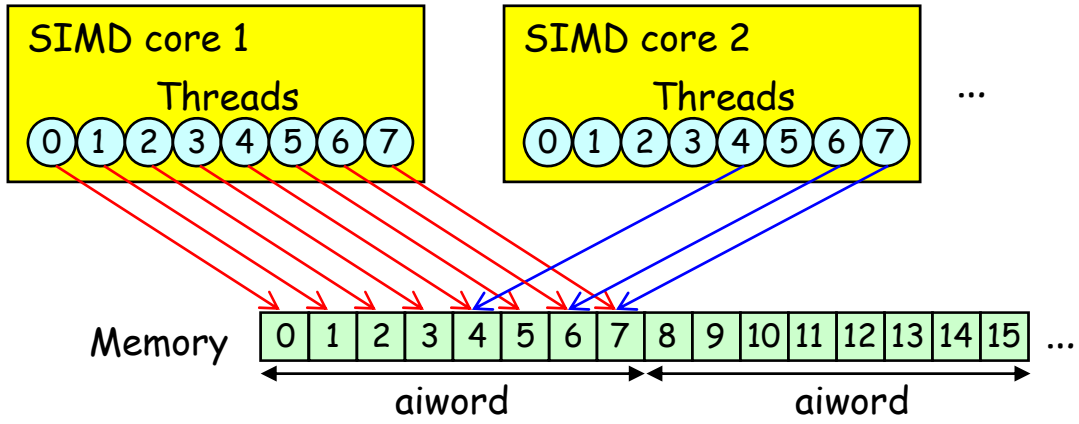
[4,5,6,7] ⇒ q cannot determine if p has written!

Key idea 2

- Some of the *n* bytes of a word may be left untouched in a single-word assignment.
  - Ex: solving 2-process consensus using 4-byte writes
  - Feasibility: NVIDIA CUDA
    - Coalesced memory accesses

Aligned-inconsecutive word model (aiword)





Our main technical contributions

- Develop general models for coalesced memory accesses.
- Prove the exact consensus numbers of these models:
  - size-varying word model (svword)
  - aligned-inconsecutive word model (aiword)
  - the combination of these two models (asvword)

Road-map

- Size-varying word model (svword)
- Aligned-inconsecutive word model (aiword)
- The combination of these two models (asvword)

Size-varying word model (svword)

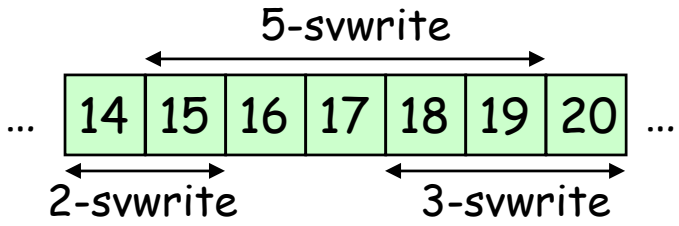
- A svword consists of b consecutive memory units,

$b \in [1,B]$, B is a constant.

  - b-svword for short
  - b-svwrite = b-svword assignment

- Alignment restriction:
  - Svwords of size b must start at addresses that are multiples of b.

- Ex: 2-svwrite, 3-svwrite and 5-svwrite

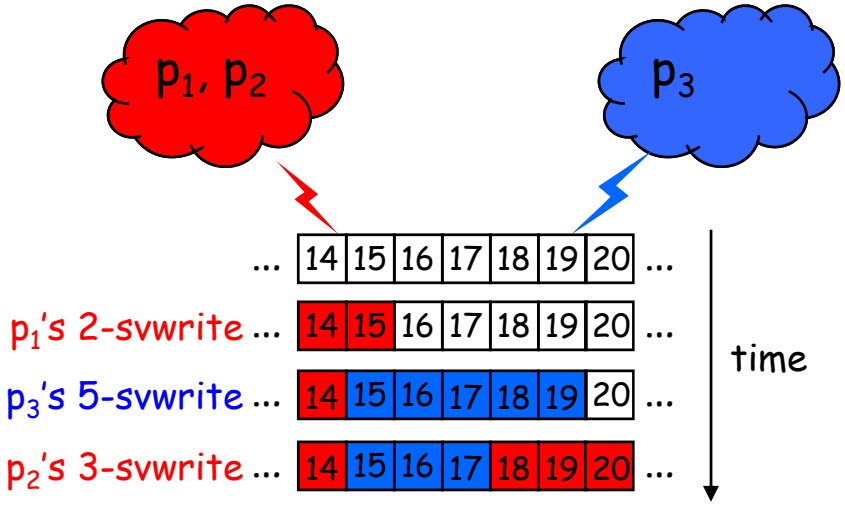


Svword's consensus no. ≥ 3

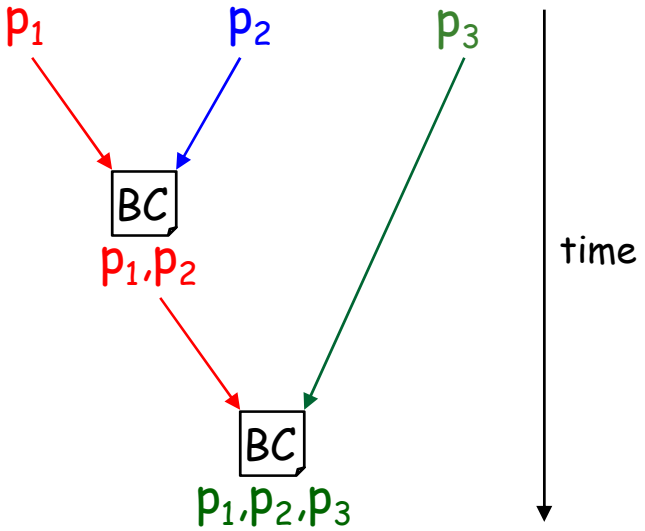
- Idea:
  - 5-svwrite can partly overlap both 2-svwrite and 3-svwrite

  ⇒ can construct (binary) consensus objects for 3 processes
- Ex:

Binary consensus (BC) for 3 processes



Consensus for 3 processes



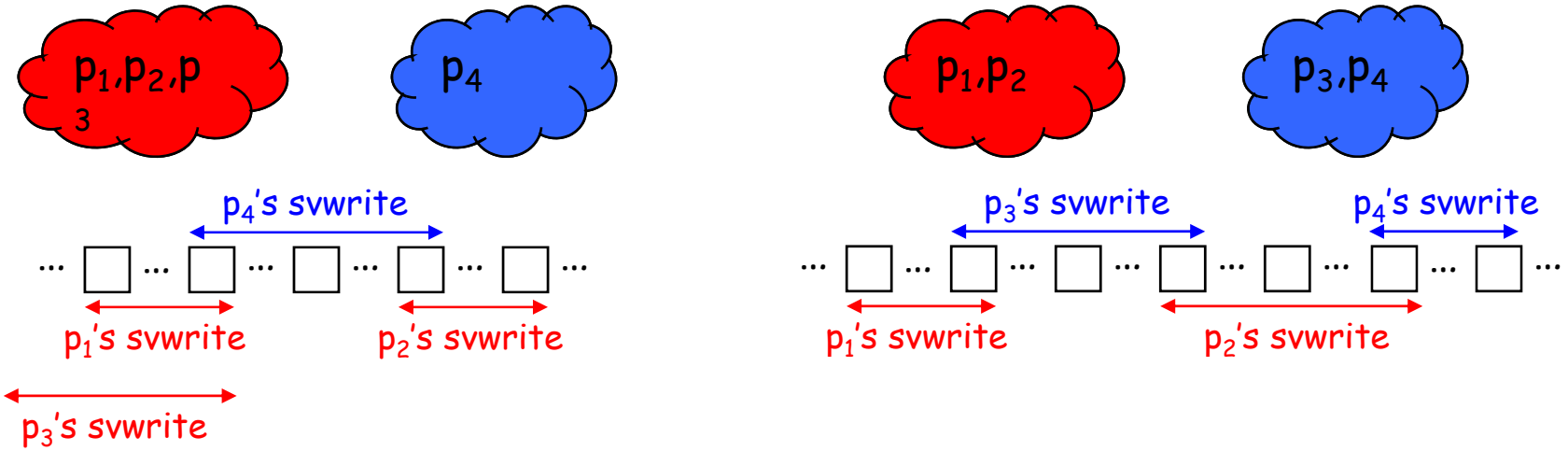
[17,18,20] ⇒ p_3's write → p_2's write

[14,15,16] ⇒ p_1's write → p_3's write ⇒ red wrote first ⇒ agree on red

Svword's consensus no. ≤ 3

- Idea
  - p's critical assignment must
    - write to p's private unit
    - partly overlap q's critical assignment if p's critical value ≠ q's critical value

    (Bivalency argument)
  - b-svwrite accesses consecutive units ⇒ each b-svwrite can partly overlap at most 2 other b-svwrites.



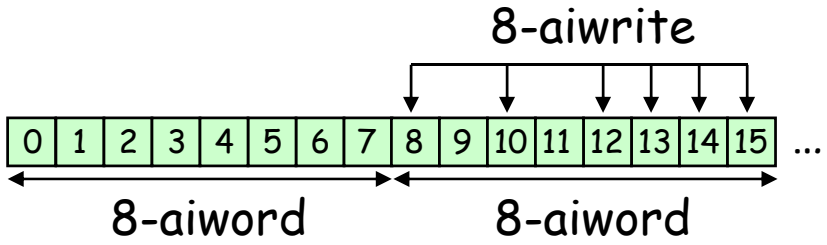
Svword's consensus number is exactly 3

Road-map

- Size-varying word model (svword)
- Aligned-inconsecutive word model (aiword)
- The combination of these two models (asvword)

Aligned-inconsecutive word (aiword)

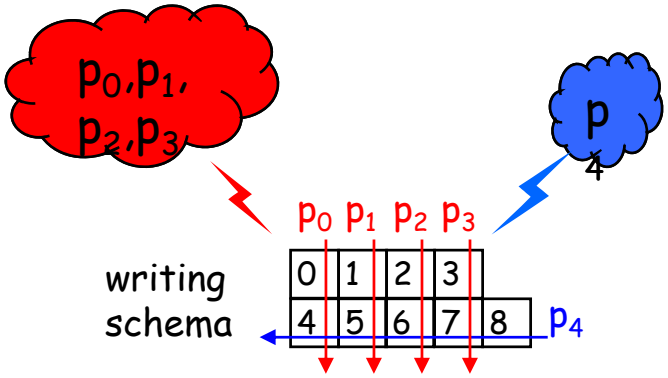
- Memory is aligned to *m*-unit words, *m* is a constant.
  - *m*-aiword for short
- A read/write operation accesses an arbitrary non-empty subset of the *m* units of an aiword.
  - *m*-aiwrite = *m*-aiword assignment.
- Alignment restriction
  - *m*-aiwords must start at addresses that are multiples of *m*.
- Ex: 8-aiwrite

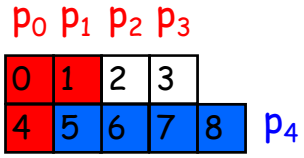


m-aiword's consensus no. ≥ ⌊(m+1)/2⌋

- Idea:
  - Construct a *binary* consensus object for $N=\lfloor (m+1)/2 \rfloor$ processes in which $(N-1)$ processes propose the same value.
  - Construct a *multivalued* consensus object for N processes using the binary consensus object.
- Ex: 9-aiword

Binary consensus (BC) for 4+1 processes

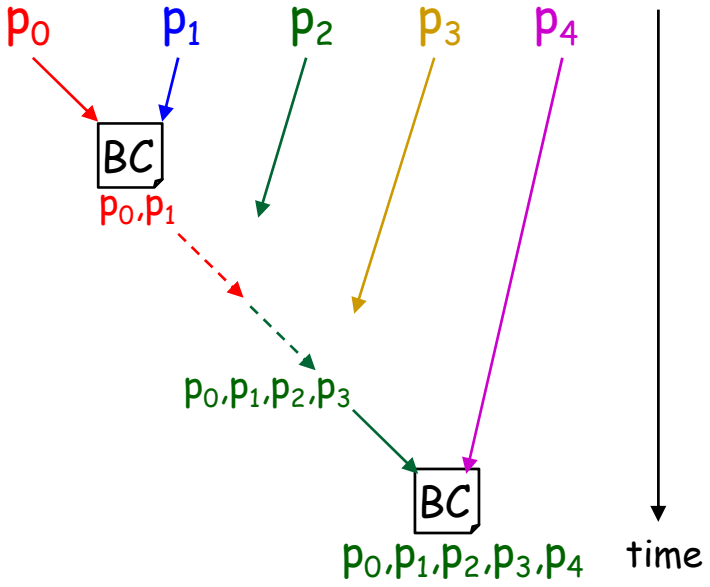




$[0,\underline{4},8] \Rightarrow p_4 \rightarrow p_0$
$[1,\underline{5},8] \Rightarrow p_1 \rightarrow p_4$
$[2,\underline{6},8] \Rightarrow p_4 \rightarrow p_2$
$[3,\underline{7},8] \Rightarrow p_4 \rightarrow p_3$

$\Rightarrow$ red wrote first

Consensus for 5 processes



m-aiword's consensus no. $\leq \lfloor (m+1)/2 \rfloor$

- Idea:
  - Lemma: p_i's critical assignment must atomically write to
    - p_i's own unit u_i
    - shared units $u_{i,j}$ written only by p_i and p_j where p_i's critical value $cv_i \neq p_j$'s critical value cv_j.

    (Bivalency argument)
  - $\Rightarrow$ solving consensus for 2 subsets S_1 and S_2, where $cv_1 \neq cv_2$ and $n_1+n_2=N$, needs to write atomically to m units, where

    $m = N + n_1 n_2 \geq 2N - 1 \Rightarrow N \leq (m+1)/2$

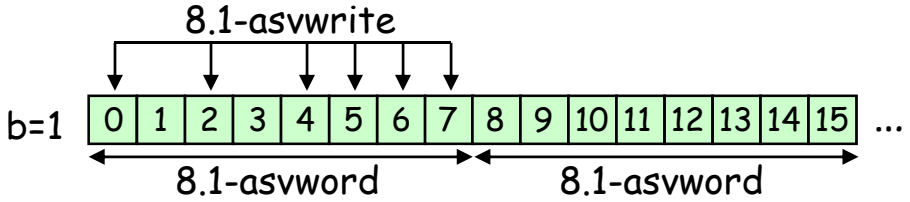
m-aiword's consensus number is exactly $\lfloor (m+1)/2 \rfloor$

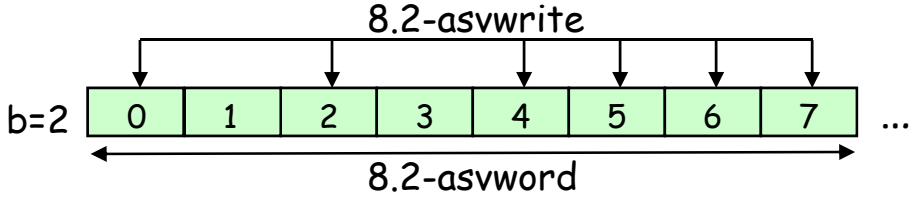
Road-map

- Size-varying word model (svword)
- Aligned-inconsecutive word model (aiword)
- The combination of these two models (asvword)

Asvword = aiword + svword

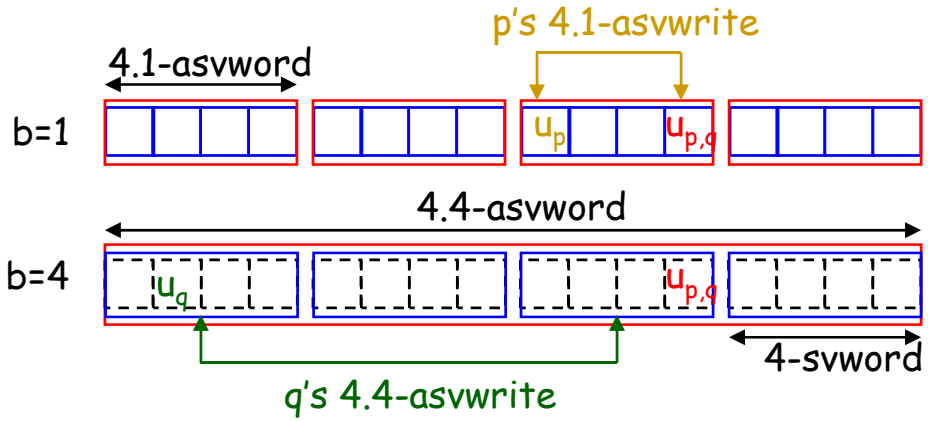
- An extension of *aiword*:
  - *aiword*'s *m* units are replaced by *m svword*s of the same size *b*, $b \in \{1,B\}$.
  - *m.b-asvword* for short
  - *m.b-asvwrite* = *m.b*-asvword assignment
  - $m=t.B$ or $B=t.m$, $t \in N^*$.
- Alignment restriction
  - *m.b*-asvwords must start at addresses that are multiples of (*m.b*).
- Ex: m=8, B=2:
  - 8.2-asvword vs. 8.1-asvword





Asvword's consensus no. when m≤B

- Asvword's consensus number is $\lfloor (m+1)/2 \rfloor$, like aiword's.
- Idea:
  - When B=t.m, $t \in N^*$, the combination of *m.1*-asvwrite and *m.B*-asvwrite does not provide any additional strength compared to *m*-aiwrite.
- Ex: B=m=4
  - p and q write to u_p, u_q, $u_{p,q}$ using 4.1-asvwrite and 4.4-asvwrite.

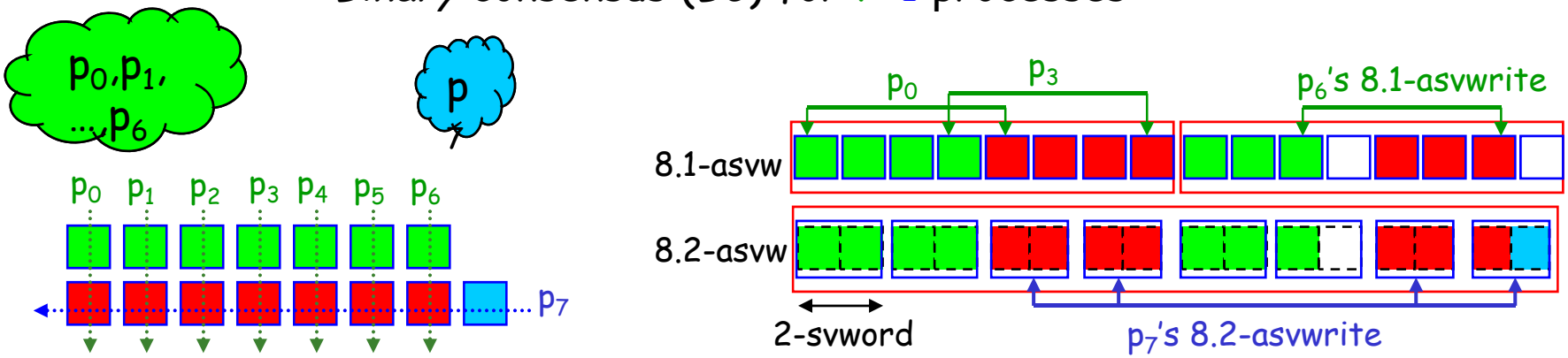


q's *4.4*-asvwrite must overwrites u_p!

Asvword's consensus no. when m>B

- Asvword's consensus number N
  - $mB/2$ if $m=2tB$, $t \in N^*$
  - $(m-B)B/2$ if $m=(2t+1)B$
- Idea
  - Processes can atomically modify $m.B$ units using $m.B$-asvwrite vs. m units using m-aiwrite.
  - Avoid overwriting unintended units:
    - each B-svword contains either private units or shared units, but not both.
- Ex: m=8, B=2 $\Rightarrow$ N=8

Binary consensus (BC) for 7+1 processes



Conclusions

- Develop new memory access models for coalesced memory accesses and prove their exact consensus numbers *N*.
  - size-varying word model, *b*-svword, $b \in [1,B]$.
    - $N = 3, \forall\ B \geq 5$
  - aligned-inconsecutive word model, *m*-aiword
    - $N = \lfloor (m+1)/2 \rfloor$
  - the combination of these two models, *m.b*-asvword, $b \in [1,B]$.

$$N = \begin{cases} \left\lfloor \frac{m+1}{2} \right\rfloor & if \quad B = tm, t \in N^* \\ \frac{mB}{2} & if \quad m = 2tB \\ \frac{(m-B)B}{2} & if \quad m = (2t+1)B \end{cases}$$

Thanks for your attention!