

## Complex stride (CPLX): Control Flow Coupled with Data Flow

Utility:

(i) The memory accesses (for a given IP) are sometimes not in the powers of two (memory layout in data structures across cache lines), causing a nonconstant stride pattern. For example, consider a cache line of 8 bytes, and if every 12th byte is accessed, the accesses create strides as follows: byte addresses: 0, 12, 36, 48, 72; cache line aligned addresses: 0, 1, 3, 4, 6; strides: 1, 2, 1, 2.

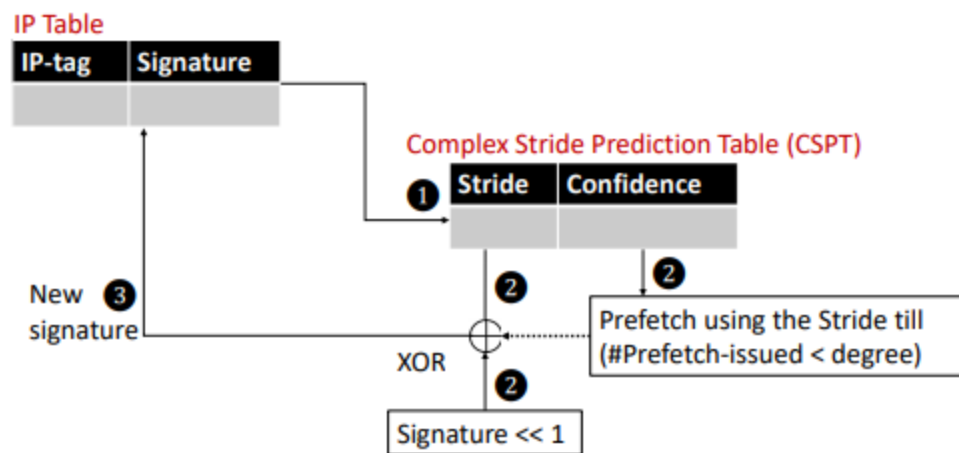
(ii) Another case is where the accesses are made by loops at various levels. An outer loop could make constant stride accesses (can be easily captured by the IP-Stride class). However, an inner loop could make different stride accesses (depending on the strides of the outer loop), thus causing bumps in the stride pattern. An IP based CPLX can exploit this pattern. A CPLX's implementation is extremely lightweight since it is an L1-D prefetcher and has the added benefit of reduced latency on the critical path of issuing a prefetch at the L1D.

We create an n-bit signature of strides seen by an IP and use it to index into a complex stride prediction table (CSPT) that predicts future complex strides. An n-bit signature captures the last n strides seen by an IP by hashing. The IP table of CPLX class is also tagged and indexed by an IP. The IP table of CPLX class stores the IP-tag and the signature that points to the previous stride(s) predicted by the IP. CSPT stores the next predicted stride pointed to by a signature and a 2-bit confidence counter.. The figure given below shows the IP table of CPLX class and the CSPT table.

**Training phase:** An IP with its signature field finds the stride at the CSPT. Every time it sees the same stride the confidence counter is incremented by one and decremented otherwise. This stride is hashed with the existing signature, and the CSPT is looked up again to issue prefetch requests. The stride obtained previously is added to the signature according to the equation:  $\text{signature} = (\text{signature} \ll 1) \oplus \text{stride}$ . Note that we shift the signature by a single bit so that we can accommodate a highly complex stride pattern. Thus a pattern can produce many signatures, but we do not observe too many collisions in the CSPT because there are not many CPLX IPs at the same point of time.

**Note:** Feel free to use any other method to store and update the signatures.

**Trained phase:** Every time the signature points to the stride, and if the confidence is high enough, the complex stride is added to the cache line to produce the prefetch address. This look-ahead continues until the prefetch degree count is reached ( 2 , 3 , and 1 ). If the confidence value is zero, then the stride is added to the signature using the above equation to predict the next stride ( 3 ) and no prefetching is done.



Credit: Samuel Pakalapati and Biswabandan Panda