



a place of mind

THE UNIVERSITY OF BRITISH COLUMBIA

VENICE: A Compact Vector Processor

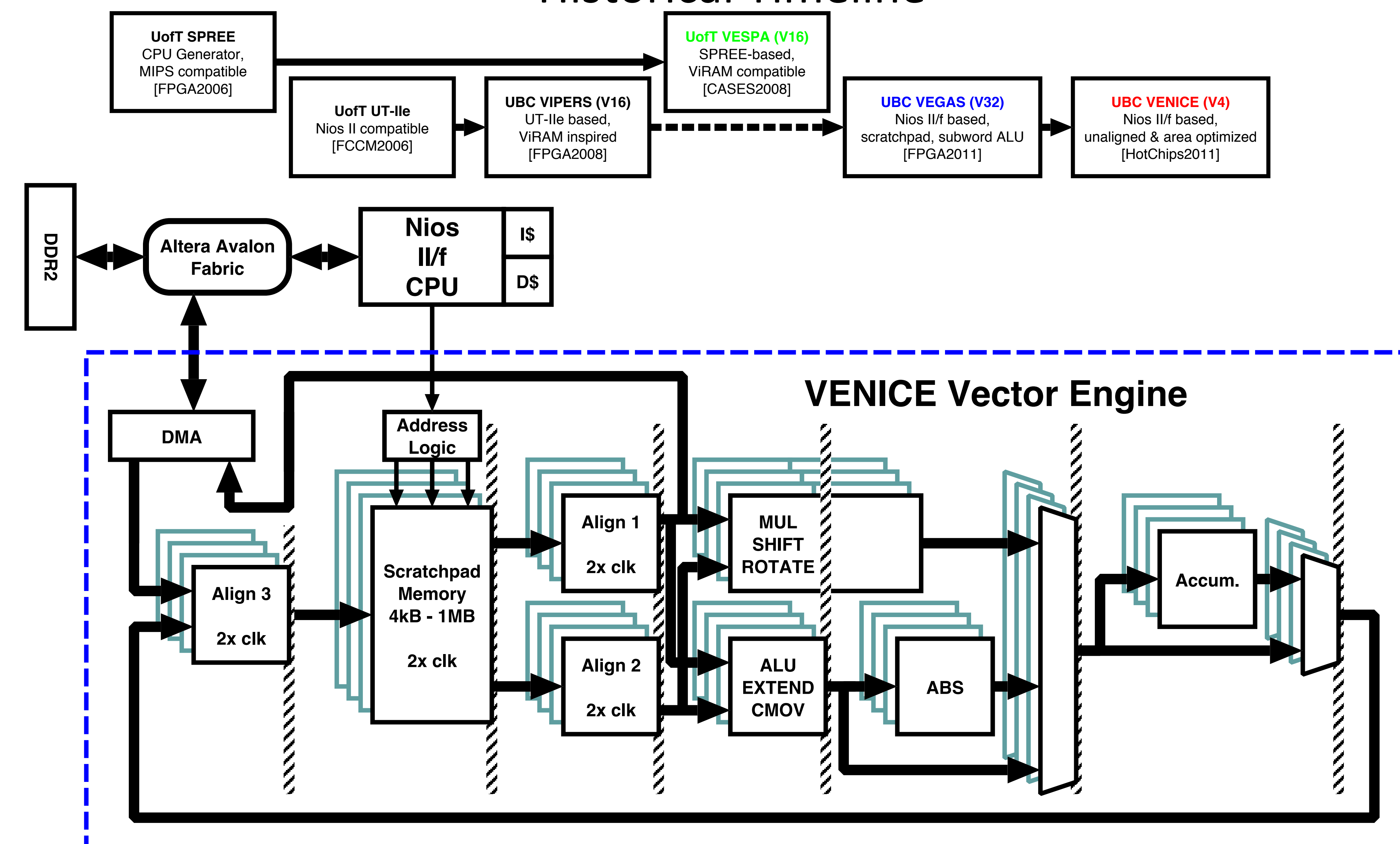


for FPGA Applications

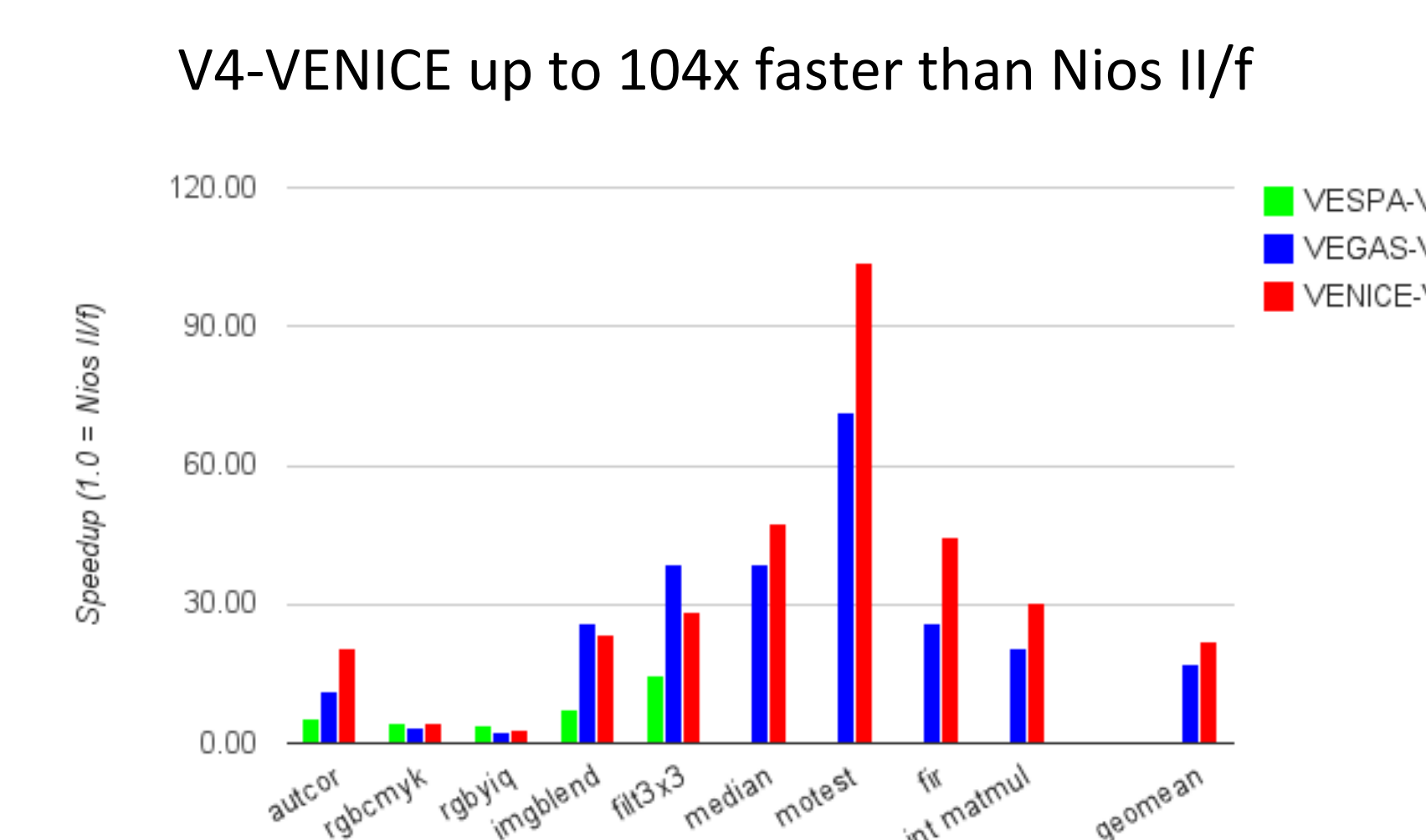
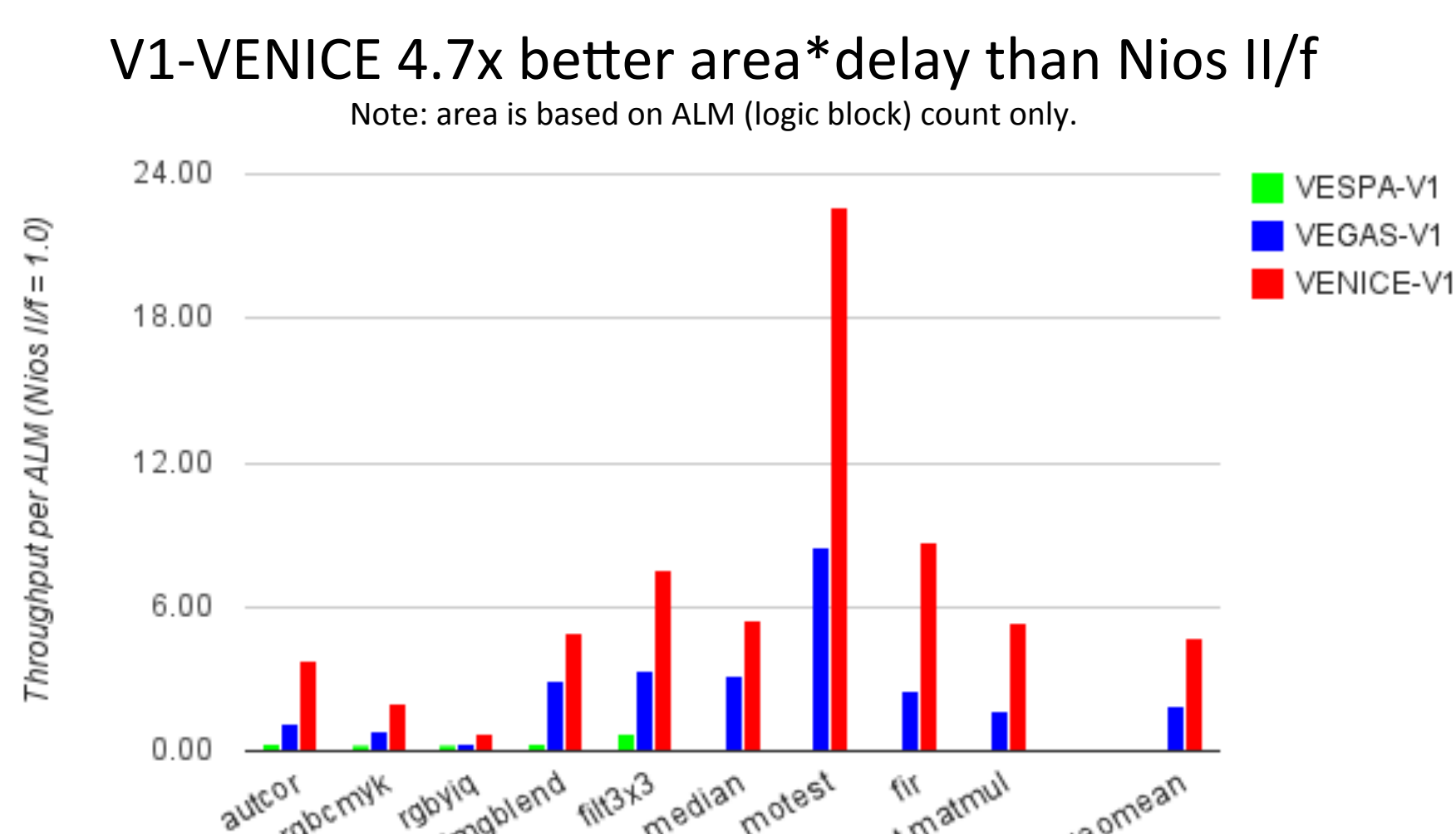
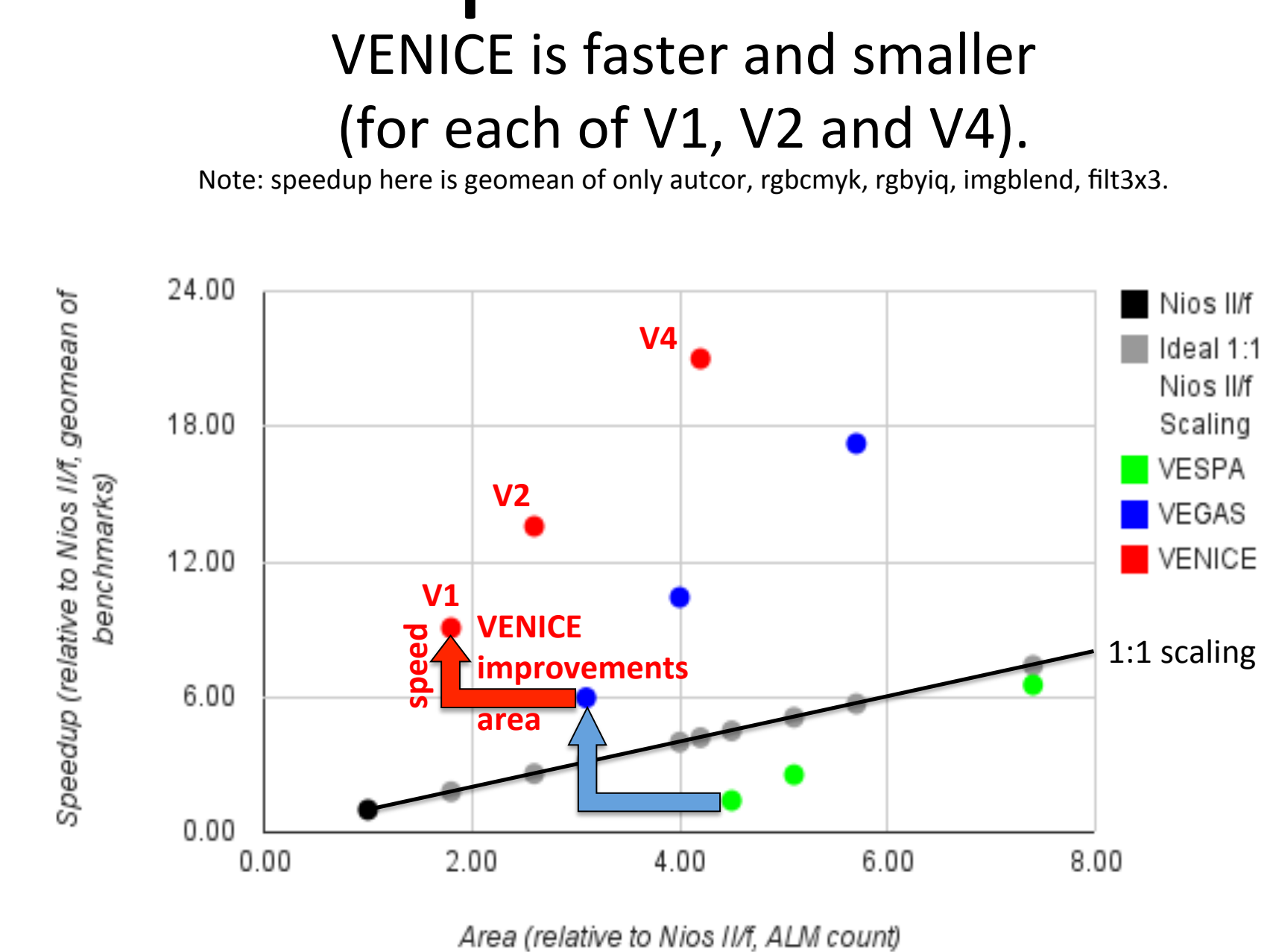
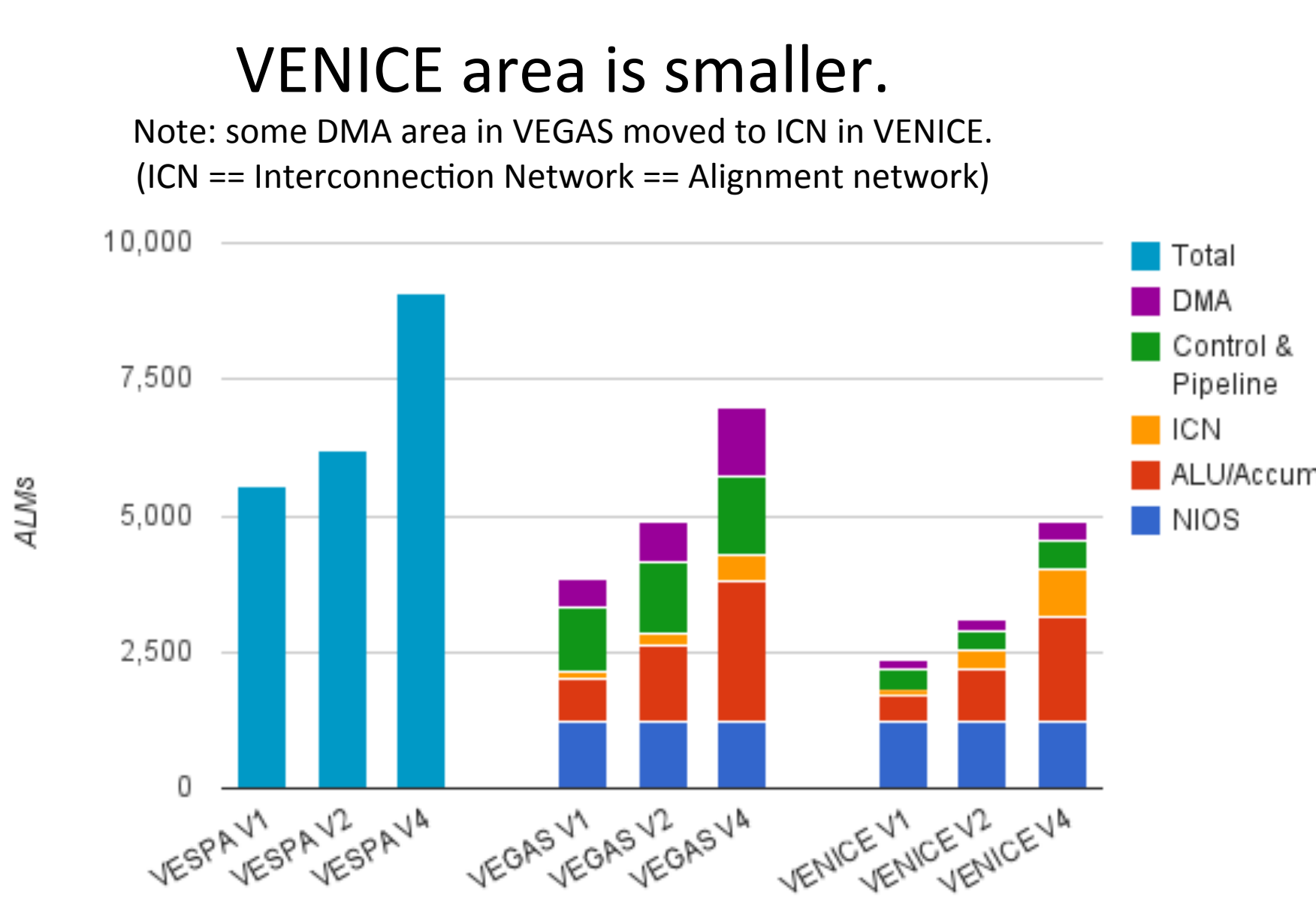
Aaron Severance

Prof. Guy Lemieux

Historical Timeline



Area and Speed Comparisons



Code Sample – Blocked and Transposed FIR

```
// The code below is single-buffered, but double-buffered is possible and will hide memory latency.
void vector_fir( uint16 *input, uint16 *output, uint16 *coeffs, int sample_size, int num_taps )
{
    // divide scratchpad into maximally-sized pieces
    // all code below assumes chunk_size < sample_size
    int chunk_size = VEGAS_VECTOR_MEMORY_SIZE / ( (3*chunk_size+4*num_taps)*sizeof(uint16) );

    // allocate memory from scratchpad
    uint16 *sample_on_vpu = (uint16 *)vector_malloc( ( chunk_size+num_taps)*sizeof(uint16) );
    uint16 *mult = (uint16 *)vector_malloc( ( chunk_size+num_taps)*sizeof(uint16) );
    uint16 *dest_on_vpu = (uint16 *)vector_malloc( (num_taps+chunk_size+num_taps)*sizeof(uint16) );

    dest_on_vpu += num_taps;

    int j, chunk_start = 0;
    while( chunk_start < sample_size ) {
        vector_dma_to_vector( sample_on_vpu, input+chunk_start, (chunk_size+num_taps)*sizeof(uint16) );
        vector_wait_for_dma();

        vector_set_vl( chunk_size+num_taps ); // sets vector length

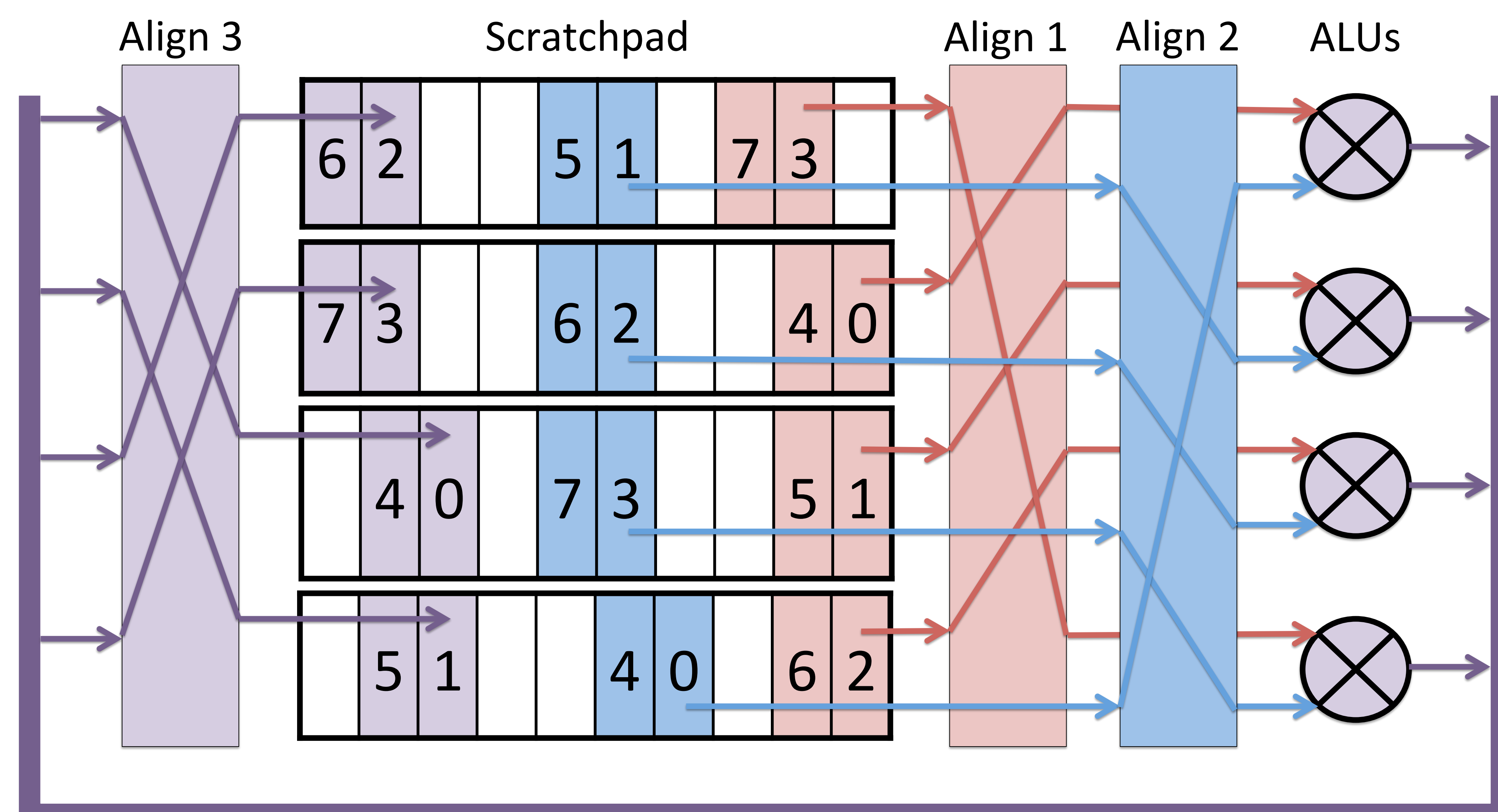
        uint16 *temp_dest = dest_on_vpu;
        vector( SVHU, VMULLO, temp_dest, coeffs[0], sample_on_vpu );
        for( j = 1; j < num_taps; j++ ) {
            temp_dest--;
            vector( SVHU, VMULLO, mult, coeffs[j], sample_on_vpu );
            vector( VVHU, VADD, temp_dest, temp_dest, mult );
        }

        vector_wait_for_dma();
        vector_instr_sync(); // waits for all vector instructions to finish

        vector_dma_to_host( output+chunk_start, dest_on_vpu, chunk_size*sizeof(uint16) );
        chunk_start += chunk_size;
    }

    vector_free(); // frees entire scratchpad
    vector_wait_for_dma();
}

```



Scratchpad supports: (1) arbitrary number of vectors, (2) any starting position, (3) any length, (4) 2D vectors (ie, submatrices).