WebAssembly SIMD

Community Group Meeting
May 2017

SIMD Availability

Width	64	128	256	512
Intel	MMX	SSE	AVX	AVX-512
ARMv7/v8	NEON	NEON		
MIPS		MSA		
POWER		VSX		

128-bit Packed SIMD

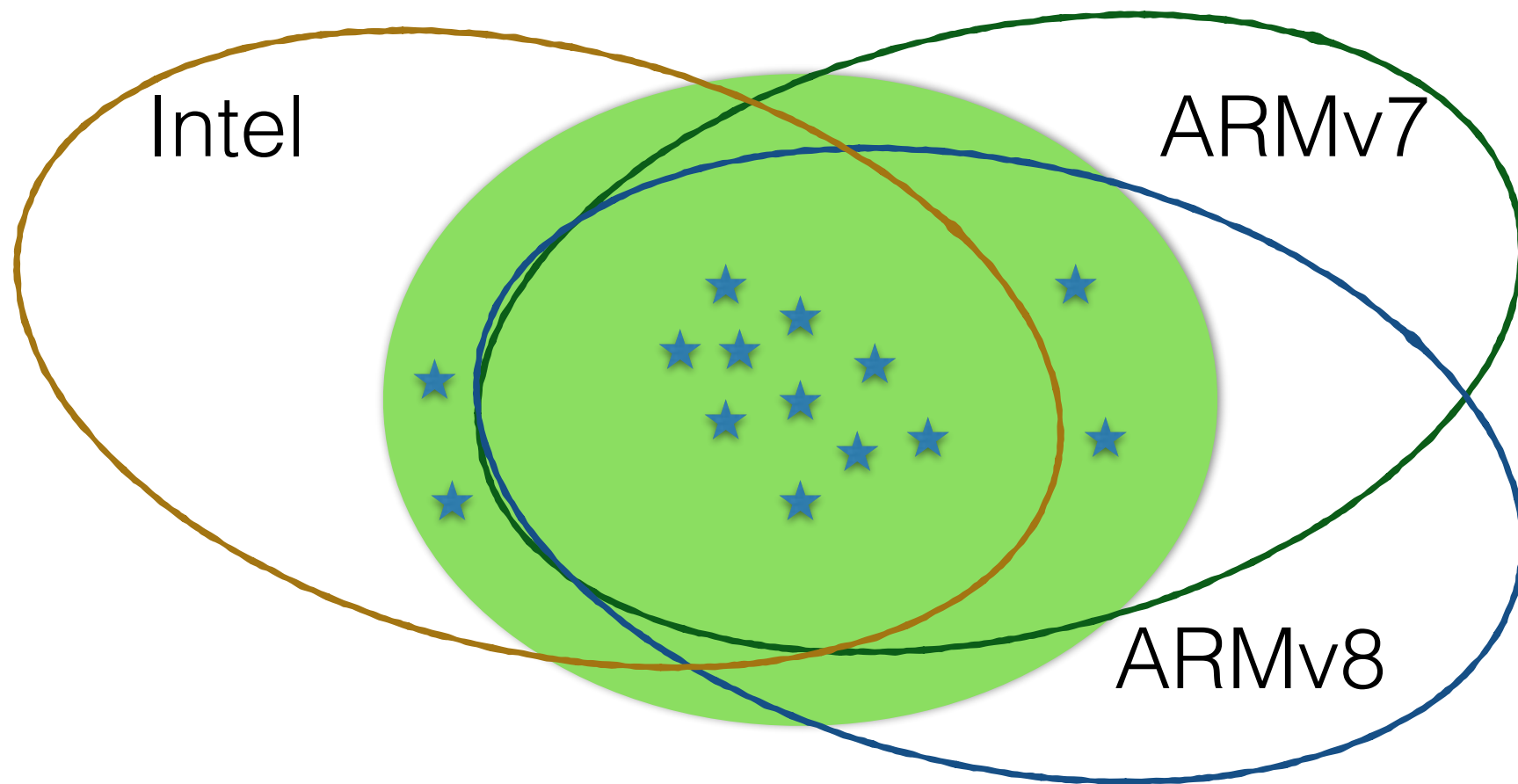| v128							
f64				f64			
i64				i64			
f32		f32		f32		f32	
i32		i32		i32		i32	
i16	i16	i16	i16	i16	i16	i16	i16
i8 i8	i8 i8	i8 i8	i8 i8	i8 i8	i8 i8	i8 i8	i8 i8
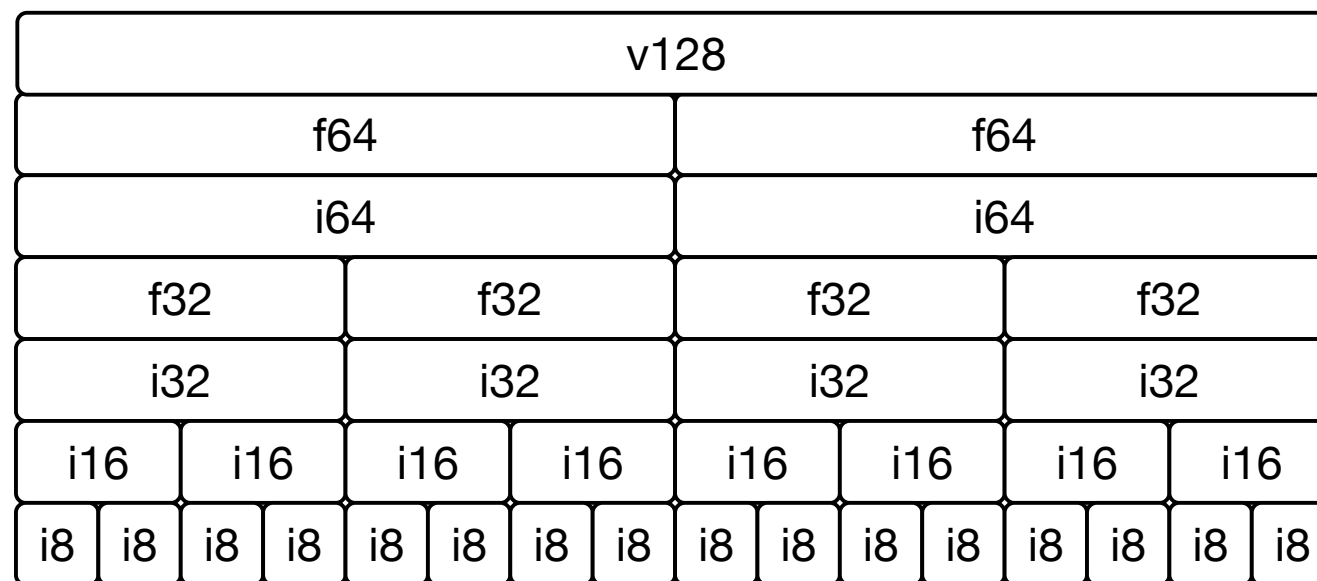
Not: GPUs, ARM SVE, Cray-style vectors

SIMD Programs

- Existing sources using <x86intrin.h> or <arm_neon.h>

- Portable code using vector extensions

- Auto-vectorizers

Portable Subset

Portable Subset

- Based on SIMD.js work

- Basics: v128.const, lane access, load/store

- Lane-wise versions of scalar instructions

- Shuffles

- No performant alternatives: Saturating arithmetic, floating-point reciprocal approximations

- More can be added in the future

Omitted

- Integer division: Not available in any ISA, requires scalar expansion anyway

- Rotates: Only in AVX-512. Use shifts + or instead

- Bit-counting instructions: Availability is spotty, applications are not obvious

- Floating point rounding instructions: Need to manipulate FP control register

- Fused multiply-add: Very expensive to emulate

Candidates

- Horizontal pairwise addition, integer and floating point

- Widening integer conversion, signed and unsigned

- Saturating narrowing integer conversion