Welcome

To Advance through Presentation
Use Page Up and Page Down Keys

Vectorizing Your Code

A. Sazegari
AltiVec Technical Lead
Core OS

Session Overview

- Detailed view of AltiVec Engine
- Inner workings of the programming model
- Underpinnings of the Libraries
- Example of AltiVec programming
- Simulation of AltiVec Performance: SimG4

Units

- Permute Cross-bar
- Vector Simple unit
- Vector Complex unit
- Vector FP unit
- Alignment and data touch

Examples

- Vector simple: vaddubm (unsigned byte modulo)
- Vector complex: vmsumshm (vector mult sum signed halfword modulo)
- Vector FP: vmaddfp (mult-add fused/fp)
- Permute: vperm
- Alignment/DST example

Key Instructions

- Pack and unpack
- Shifts and rotates
- Permute and select
- Merge and splat
- FP rounds and estimates

Inner Workings of the Programming Model

- Data Types
- Coercion and Promotion
- Generic Operator Type Sensitivity
- Function Call Parsing

Libraries

- vBasicOps
  complete architecture of basic ops
  (+-x/ up to 128 bit signed and unsigned
  modulo and saturated)
- vectorOps
  vector versions of matrix operations
- vMathLib
  IEEE754-compliant auxiliary,
  trancendental and special functions

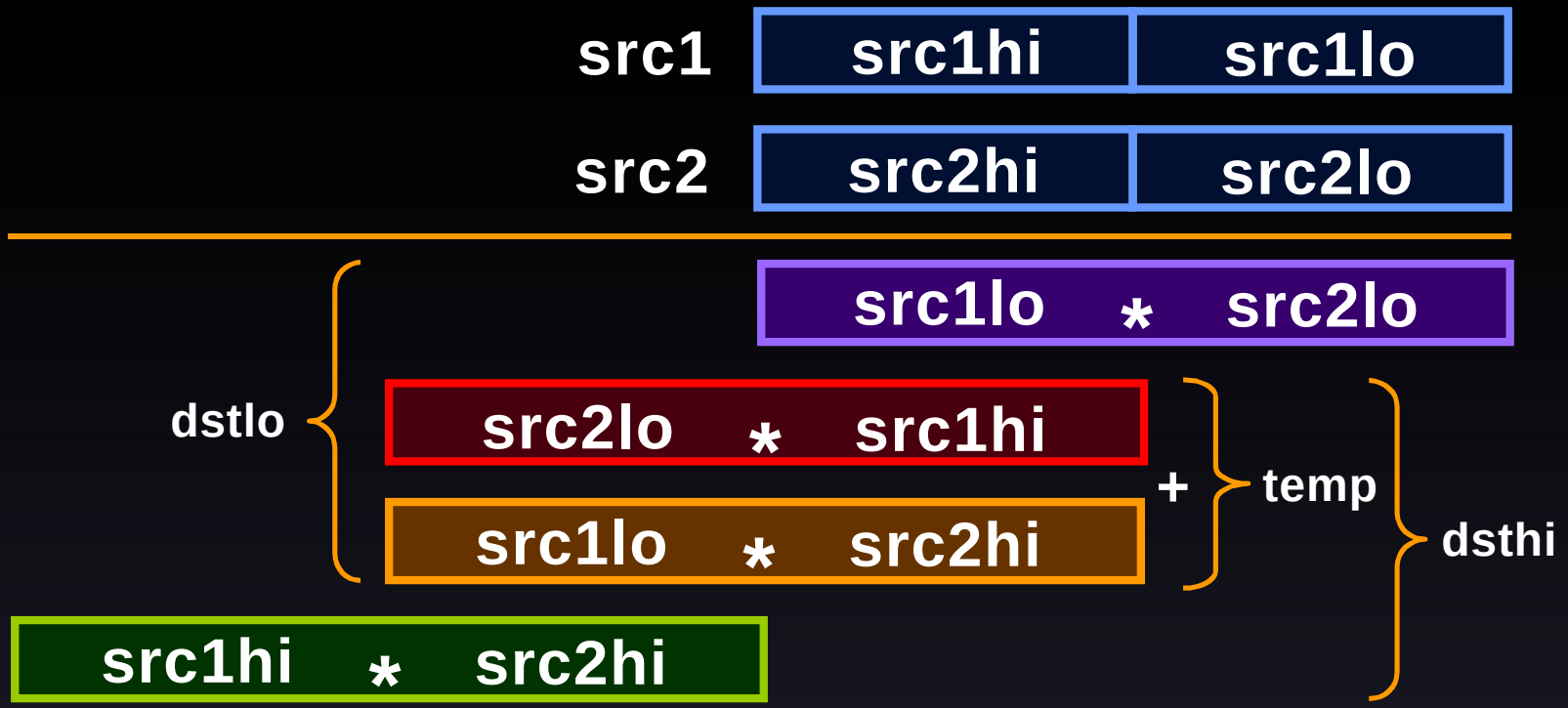
Libraries (Cont.)

- vBigNum
  large-number arithmetic up to 1024 bits
- vDSPLib
  fft, dft, windowing and squaring
- vImageLib
  translation, rotation, reversals, etc.
- bigDspLib
  large FFT and large wavelets

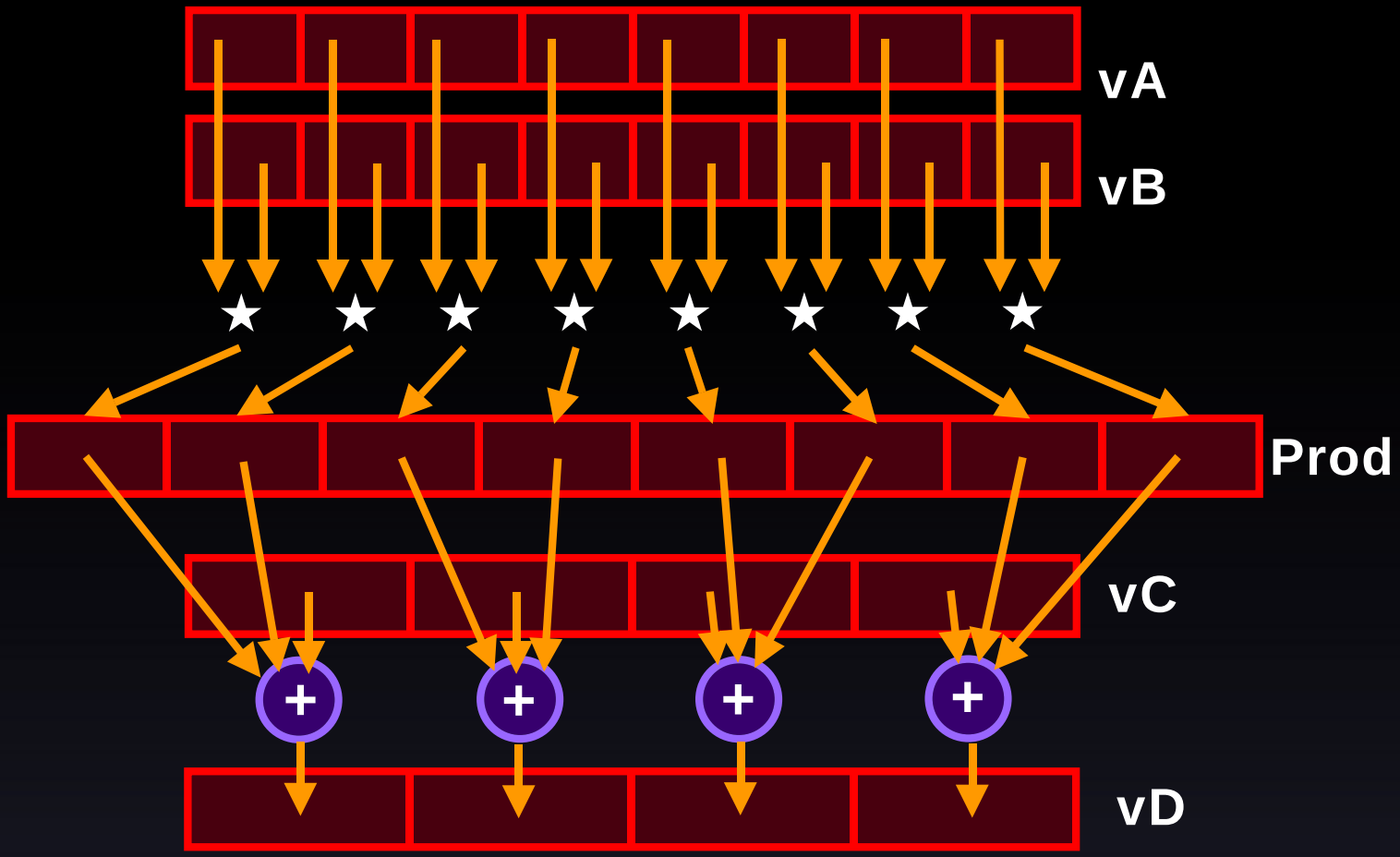
Example of AltiVec Programming

- Multiply example comparison of vector and scalar code, section by section for 32x32bit full multiply

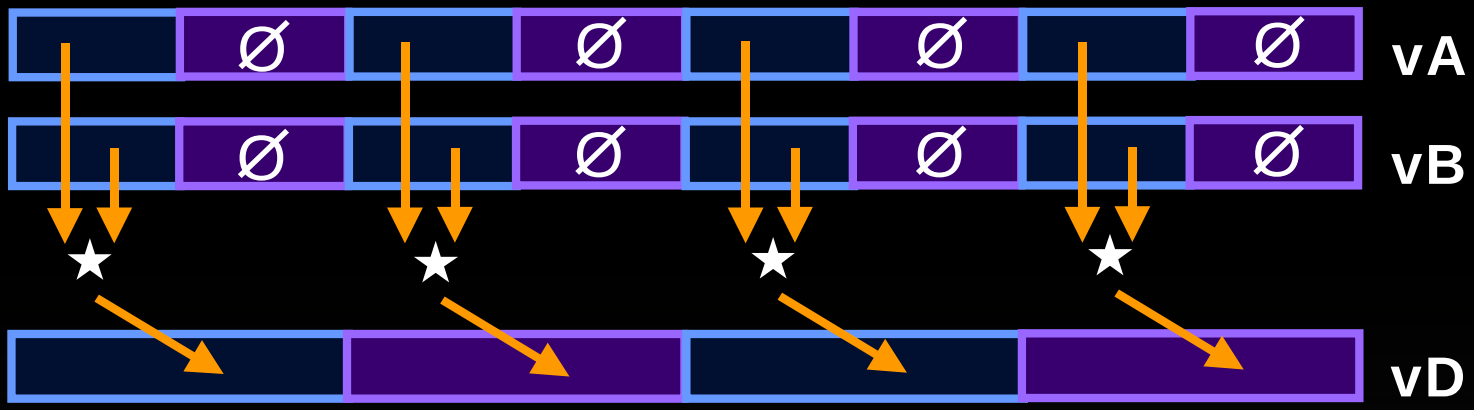
Full 32-bit Multiply



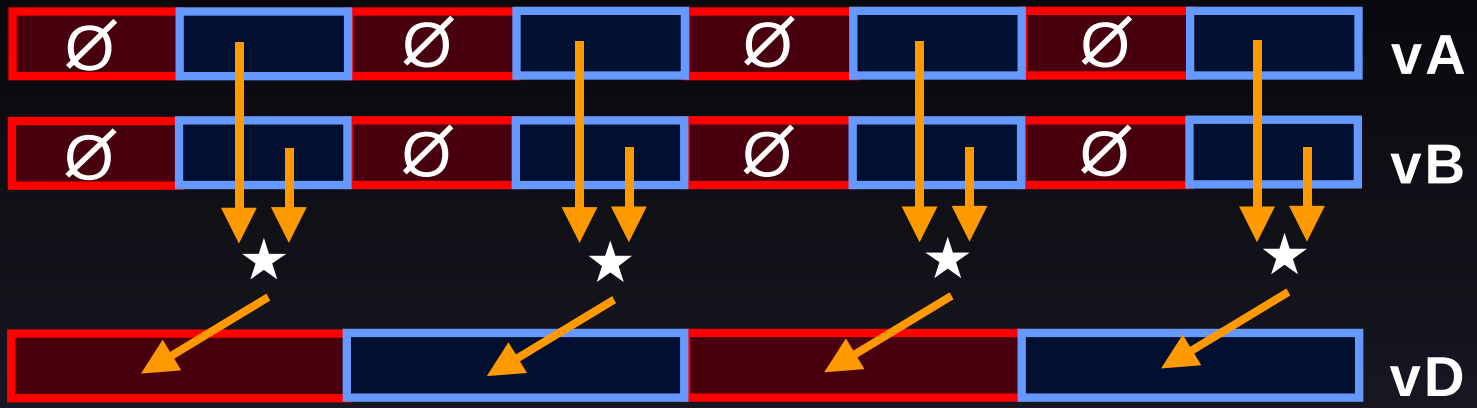
Carry: if (dstlo < temp)
dsthi +=1



Vmsumuhm—Multiply-Sum of Unsigned Integer Elements (16-Bit to 32-Bit)



vmuleuh—Even Multiply of Four Unsigned Integer Elements (16-Bit)



vmulouh—Odd Multiply of Four Unsigned Integer Elements (16-Bit)

altivec™ technology

Douglas Clarke

AltiVec Vectorizer

The Permute Unit

- What it does
- Every Other Sample
- Merging Streams
- Aligning Data

The Permute Unit

- Creates a vector by arbitrarily selecting 16 bytes from two 16 byte input vectors
- Can be controlled by an input control vector
- Can be controlled by an opcode implied vector

Every Other Sample

- Given a stream of data (longs), how can we reduce the stream by 50% (take every other sample)
- By using the permute unit, we can read in 8 values from the stream and from them create a vector that contains 4 values

Every Other Sample

```
For ( I=0; I < streamLength; I+=2)
{
   v1 = *inPtr++;
   v2 = *inPtr++;

   vOut = vec_pack ( v1, v2 );

   *outPtr++ = vOut;
}
```

Every Other Sample

```
For ( I=0; I < streamLength; I+=2)
{
   v1 = *inPtr++;
   v2 = *inPtr++;

   vOut = vec_pack ( v1, v2 );

   *outPtr++ = vOut;
}
```

Every Other Sample

```
For ( I=0; I < streamLength; I+=2)
{
   v1 = *inPtr++;
   v2 = *inPtr++;

   vOut = vec_pack ( v1, v2 );

   *outPtr++ = vOut;
}
```

Every Other Sample

```
For ( I=0; I < streamLength; I+=2)
{
   v1 = *inPtr++;
   v2 = *inPtr++;

   vOut = vec_pack ( v1, v2 );

   *outPtr++ = vOut;
}
```

Merging Streams

- Given two stream of data (shorts), how can we merge the streams into one stream with alternating shorts from each source?

Merging Streams

```
For ( I=0; I < streamLength; I ++)
{
   v1 = *inPtr++;
   v2 = *inPtr++;

   *outPtr++ = vec_mergeh ( v1, v2);
   *outPtr++ = vec_mergel ( v1, v2);
}
```

Merging Streams

```
For ( I=0; I < streamLength; I ++)
{
   v1 = *inPtr++;
   v2 = *inPtr++;

   *outPtr++ = vec_mergeh ( v1, v2);
   *outPtr++ = vec_mergel ( v1, v2);
}
```

Merging Streams

```
For ( I=0; I < streamLength; I ++)
{
   v1 = *inPtr++;
   v2 = *inPtr++;

   *outPtr++ = vec_mergeh ( v1, v2);
   *outPtr++ = vec_mergel ( v1, v2);
}
```

Merging Streams

```
For ( I=0; I < streamLength; I ++)
{
   v1 = *inPtr++;
   v2 = *inPtr++;

   *outPtr++ = vec_mergeh ( v1, v2);
   *outPtr++ = vec_mergel ( v1, v2);
}
```

Merging Streams

```
For ( I=0; I < streamLength; I ++)
{
   v1 = *inPtr++;
   v2 = *inPtr++;

   *outPtr++ = vec_mergeh ( v1, v2);
   *outPtr++ = vec_mergel ( v1, v2);
}
```

Aligning Data

- Given a stream of data (16 byte), how can we work on it starting from an arbitrary offset ?
- The permute can be used to align data
- For runtime offsets we can permute on the fly

Aligning Data

```
Align_vector = vec_lvsl ( offset, 0);
v1 = *inPtr++;
For ( I=0; I < streamLength; I ++)
{
   v2 = *inPtr++;

   vOut = vec_perm ( v1, v2, align_vector );

   *outPtr++ = vOut;
   v1 = v2;
}
```

Aligning Data

```
av = vec_lvsl ( offset, 0);
v1 = *inPtr++;
For ( I=0; I < streamLength; I ++)
{
   v2 = *inPtr++;

   vOut = vec_perm ( v1, v2, align_vector );

   *outPtr++ = vOut;
   v1 = v2;
}
```

Aligning Data

```
av = vec_lvsl ( offset, 0);
v1 = *inPtr++;
For ( I=0; I < streamLength; I ++)
{
   v2 = *inPtr++;

   vOut = vec_perm ( v1, v2, align_vector );

   *outPtr++ = vOut;
   v1 = v2;
}
```

Aligning Data

```
av = vec_lvsl ( offset, 0);
v1 = *inPtr++;
For ( I=0; I < streamLength; I ++)
{
   v2 = *inPtr++;

   vOut = vec_perm ( v1, v2, align_vector );

   *outPtr++ = vOut;
   v1 = v2;
}
```

Aligning Data

```
av = vec_lvsl ( offset, 0);
v1 = *inPtr++;
For ( I=0; I < streamLength; I ++)
{
   v2 = *inPtr++;

   vOut = vec_perm ( v1, v2, align_vector );

   *outPtr++ = vOut;
   v1 = v2;
}
```

Aligning Data

```
av = vec_lvsl ( offset, 0);
v1 = *inPtr++;
For ( I=0; I < streamLength; I ++)
{
   v2 = *inPtr++;

   vOut = vec_perm ( v1, v2, align_vector );

   *outPtr++ = vOut;
   v1 = v2;
}
```

Call to Action

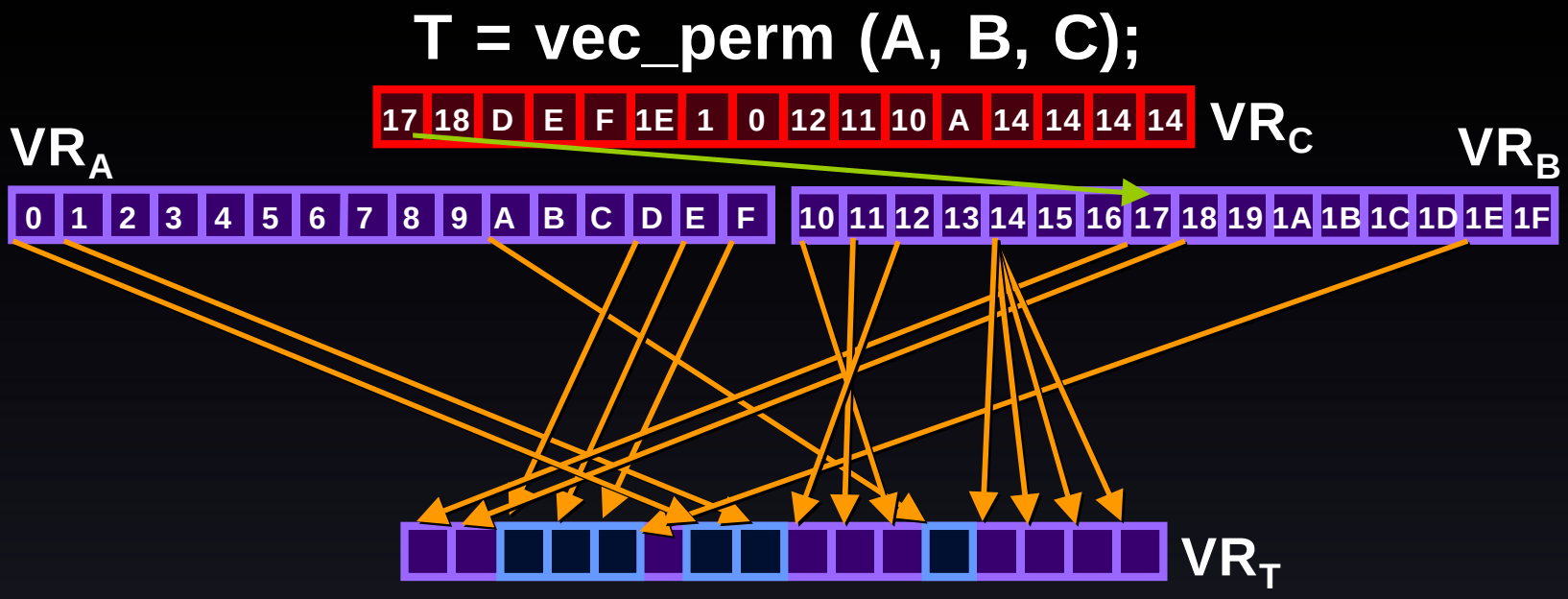
- Attend other AltiVec sessions
- Dowload the SDK:
  <developer.apple.com/hardware/altivec>
- Identify data parallelism in your programs
- Vectorize critical sections first without DST
- Add DST instructions last
- Use SIM_G4 to tune performance
- Sign up for the next AltiVec kitchen

altivec™ technology

Douglas Clarke

AltiVec Vectorizer

Vector Permute



- Arbitrary bytewise data reorganization
- Small table-lookup

Every Other Sample

Every Other Sample

The Data
12 13 20 40 50 60 70 80

Every Other Sample

The Data
12 13 20 40 50 60 70 80

V1 = 12 13 20 40

Every Other Sample

The Data

12 13 20 40 **50 60 70 80**

V1 = 12 13 20 40

V2 = 50 60 70 80

Every Other Sample

The Data
12 13 20 40 50 60 70 80

V1 = 12 13 20 40
V2 = 50 60 70 80

vOut = 13 40 60 80

Merging Streams

Merging Streams

The Data

Stream1 10 20 30 40 50 60 70 80

Stream2 55 56 57 58 59 60 61 62

Merging Streams

The Data
Stream1 10 20 30 40 50 60 70 80
Stream2 55 56 57 58 59 60 61 62

v1 = 10 20 30 40 50 60 70 80

Merging Streams

The Data

Stream1 10 20 30 40 50 60 70 80

Stream2 55 56 57 58 59 60 61 62

v1 = 10 20 30 40 50 60 70 80

v2 = 55 56 57 58 59 60 61 62

Merging Streams

The Data
Stream1 10 20 30 40 50 60 70 80
Stream2 55 56 57 58 59 60 61 62

v1 = 10 20 30 40 50 60 70 80
v2 = 55 56 57 58 59 60 61 62

vOut= 10 55 20 56 30 57 40 58

Merging Streams

The Data
Stream1 10 20 30 40 50 60 70 80
Stream2 55 56 57 58 59 60 61 62

v1 = 10 20 30 40 50 60 70 80
v2 = 55 56 57 58 59 60 61 62

vOut= 50 59 60 60 70 61 80 62

Aligning Data

Aligning Data

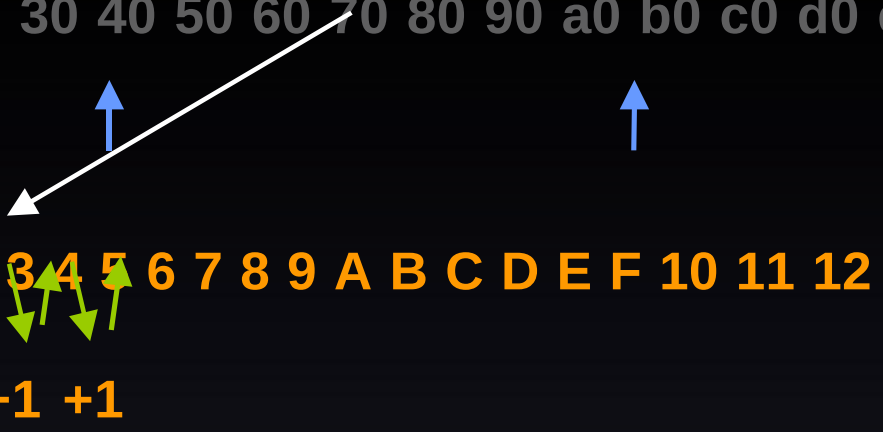
The Data offset = 3

10 20 30 40 50 60 70 80 90 a0 b0 c0 d0 e0 f0 ff

Aligning Data

The Data offset = 3

10 20 30 40 50 60 70 80 90 a0 b0 c0 d0 e0 f0 ff



Aligning Data

The Data offset = 3

10 20 30 40 50 60 70 80 90 a0 b0 c0 d0 e0 f0 ff

av = 3 4 5 6 7 8 9 A B C D E F 10 11 12

v1 = 10 20 30 40 50 60 70 80

Aligning Data

The Data offset = 3

10 20 30 40 50 60 70 80 **90 a0 b0 c0 d0 e0 f0 ff**

av = 3 4 5 6 7 8 9 A B C D E F 10 11 12

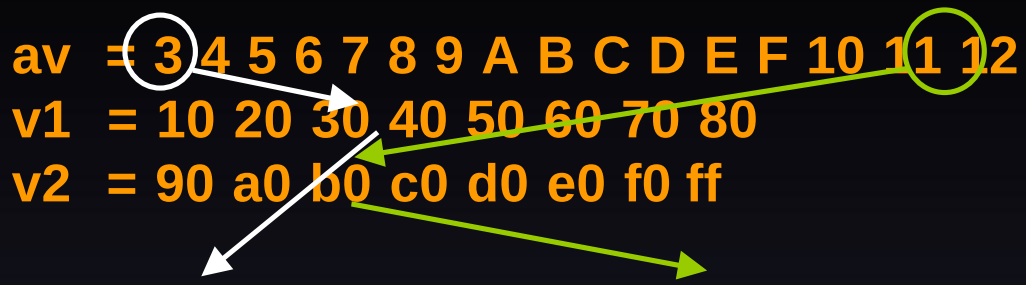
v1 = 10 20 30 40 50 60 70 80

v2 = 90 a0 b0 c0 d0 e0 f0 ff

Aligning Data

The Data offset = 3

10 20 30 40 50 60 70 80 90 a0 b0 c0 d0 e0 f0 ff



Aligning Data

The Data offset = 3

10 20 30 40 50 60 70 80 **90 a0 b0 c0 d0 e0 f0 ff**

av = 3 4 5 6 7 8 9 A B C D E F 10 11 12

v1 = 90 a0 b0 c0 d0 e0 f0 ff

More AltiVec Sessions

201: AltiVec Overview (Repeat) Send your friends	Hall 2 **Thur., 9:00am**
211: Writing AltiVec Code Performance and data streaming	Hall A1 **Fri., 9:00am**
911: AltiVec Feedback Forum Feedback on tools, support, program	Hall J2 **Fri., 10:15am**
AltiVec Workshops 2+ hours of hands-on with AltiVec	Hall L **Sign up now!**

Q&A

Helder Ramalho
Alex Rosenberg
Doug Clarke
Behzad Razban
Ali Sazegari
William Chen
Robert Murley

Think different.™

Welcome

To Advance through Presentation
Use Page Up and Page Down Keys