Advanced Optimization Techniques

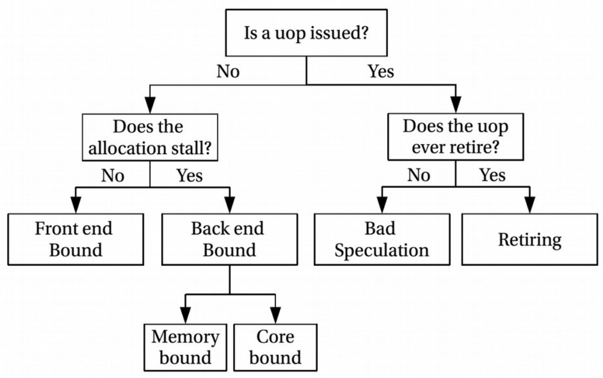
ICTP Trieste 2014

Dr. Christopher Dahnken

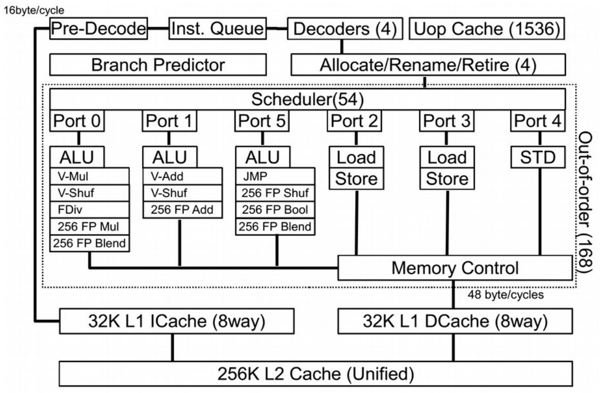
Intel GmbH

Outline

Method



CPU



Code

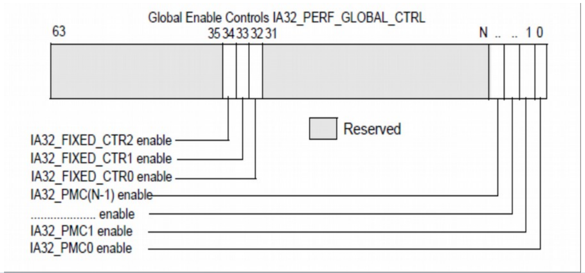
```
!$OMP SECTION
!   tsend=dclock()
 if(iblock.lt.(nblocks)) then
    nxti=m_of_i(iblock+1)
    nxtj=n_of_i(iblock+1)
    nxtk=k_of_i(iblock+1)

    nxt_buffsize_m=buffersize(ms,bm,nxti)
    nxt_index_m=(nxti-1)*bm+1

    nxt_buffsize_n=buffersize(ns,bn,nxtj)
    nxt_index_n=(nxtj-1)*bn+1

    nxt_buffsize_k=buffersize(ks,bk,nxtk)
    nxt_index_k=(nxtk-1)*bk+1
```

Measurement



Vectorization: AVX Programming Exercises

Problem 1 – Cyclic Rotate (medium)

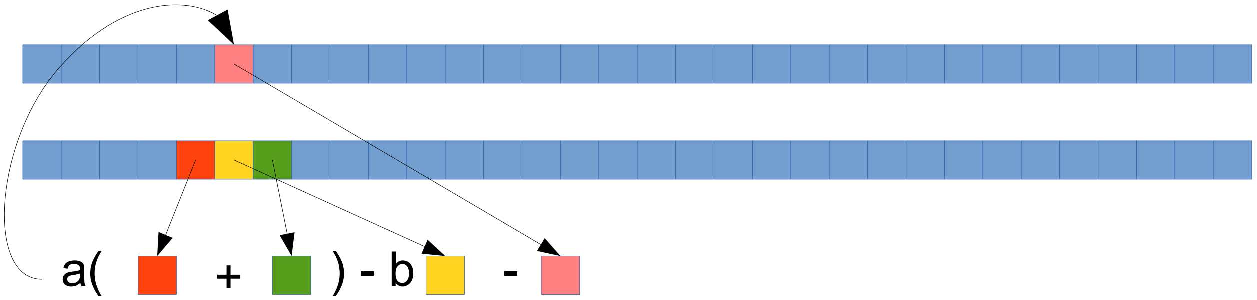
- Write a cyclic rotate in intrinsics. The function should perform the following vector operation:
- __m256d cyclic_rotate_right(__m256 a)
- Source a: (a3,a2,a1,a0)
- Result: (a0,a3,a2,a1)

Problem 2 - Wave Equation

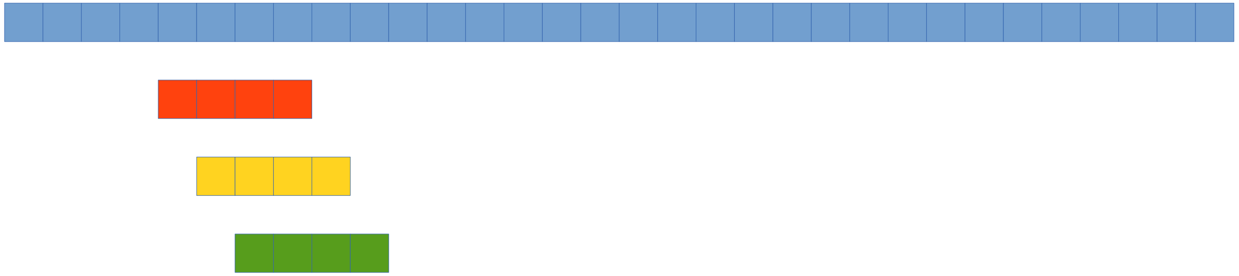
- You considered the second order finite difference solution to a 1D acoustic wave equation earlier.
- Port the C-code to AVX intrinsics
- How do you compare vs Array Notations?
- How do you compare vs normal c-code?

Problem 2 - Wave Equation

$$\varphi_x^{t+1} = a\left(\varphi_{x+1}^{t} + \varphi_{x-1}^{t}\right) - b\,\varphi_x^{t} - \varphi_x^{t-1}$$



Can you implement this so that 4 points are updated at the same time? E.g:



Problem 3 - Matrix multiplication (hard)

- Consider again a 4x4 matrix multiplication

  $$\boldsymbol{C} = \boldsymbol{C} + \boldsymbol{A} \cdot \boldsymbol{B}$$

  for a large number of matrices
- Code for a C version is supplied
- Write the matrix multiplication in AVX intrinsics
- How do you compare to compiler and AN?

Problem 3 - Matrix multiplication (hard)

- Tip: Consider the following:

$$C_{i,[0-3]} = C_{i,[0-3]} + \sum_{k} A_{i,k} \cdot B_{k,[0-3]}$$

j turns into a range

No j here – this is a vector containing four times the same element