

Automatic Throughput and Critical Path Analysis of x86 and ARM Assembly Kernels

J. Laukemann, J. Hammer, G. Hager, G. Wellein





Overview



Analytic Performance Modeling

Why?

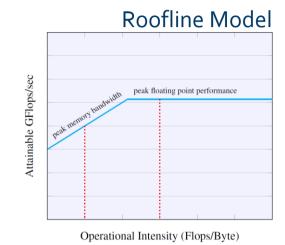
Assumptions & Related Tools

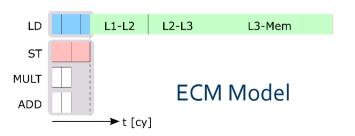
- 2. Throughput & Latency Nomenclature
 Definition of Throughput, Critical Path and Loop-Carried Dependency
- 3. OSACA: Automating the in-core model construction Overview, Structure and Output
- 4. Gauss-Seidel-Method Example
- 5. Future Work

Performance Modeling for Loop Kernels



- How fast can my kernel run at best?
- What are the relevant hardware bottlenecks?
- Apply simplified model of underlying hardware
 - In-core execution
 - Data transfer
 - Combining execution and data transfer

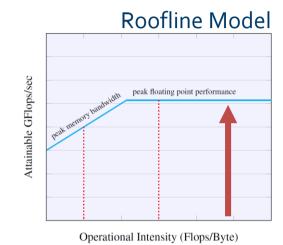




Performance Modeling for Loop Kernels



- How fast can my kernel run at best?
- What are the relevant hardware bottlenecks?
- Apply simplified model of underlying hardware
 - In-core execution
 - Data transfer
 - Combining execution and data transfer





Assumptions & Related Tools



1. All Data in L1

2. Average distribution of port scheduling

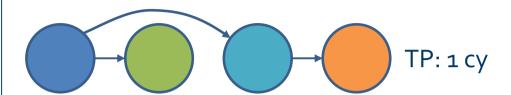
	OSACA vo.21	OSACA vo.3	IACA² (EoL)	LLVM-MCA ³
Throughput	~	~	~	~
Critical Path	X	~	X	(<u>></u>
Loop-Carried Dependencies	X	✓	X	

¹ Presented at PMBS₁₈

² Intel Architecture Code Analyzer (https://software.intel.com/en-us/articles/intel-architecture-code-analyzer)

³LLVM Machine Code Analyzer (https://llvm.org/docs/CommandGuide/llvm-mca.html)



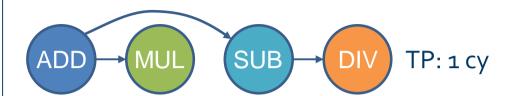


- Dependencies within loop
- No loop-carried dependencies

TP: Throughput

CP: Critical Path



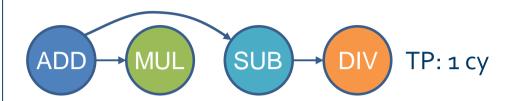


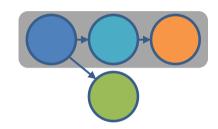
- Dependencies within loop
- No loop-carried dependencies

TP: Throughput

CP: Critical Path





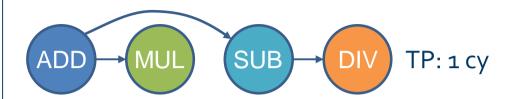


- Dependencies within loop
- No loop-carried dependencies

TP: Throughput

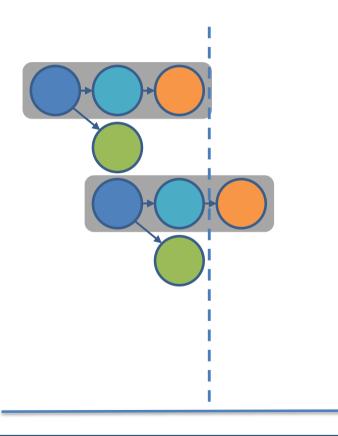
CP: Critical Path





- Dependencies within loop
- No loop-carried dependencies

TP: Throughput CP: Critical Path

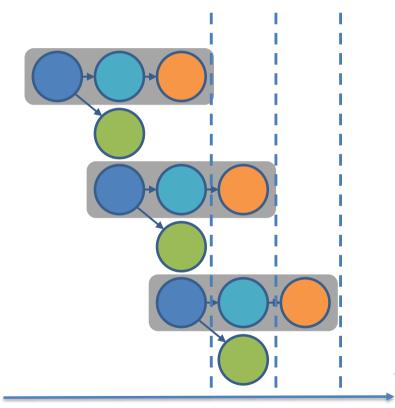




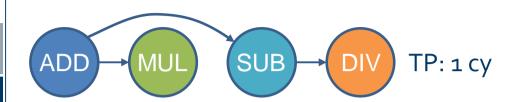


- Dependencies within loop
- No loop-carried dependencies

TP: Throughput CP: Critical Path

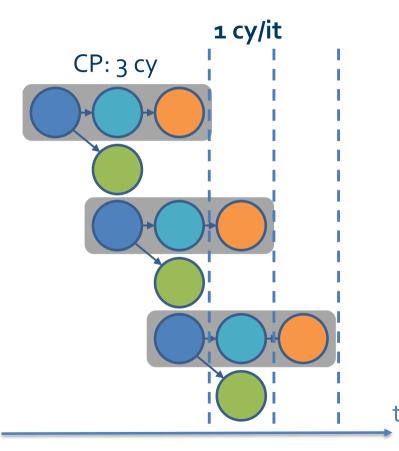




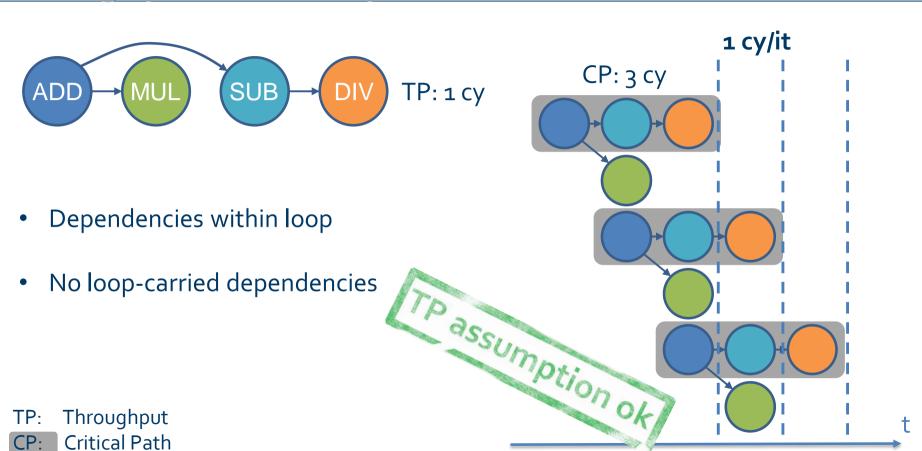


- Dependencies within loop
- No loop-carried dependencies

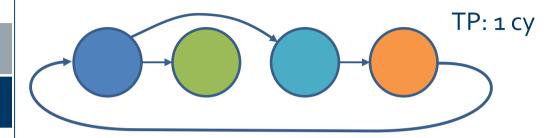
TP: Throughput CP: Critical Path









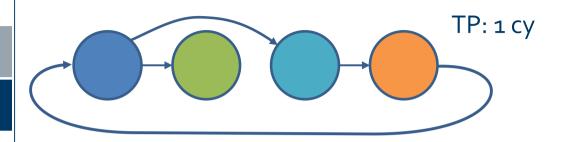


- Dependencies within loop
- Loop-carried dependencies

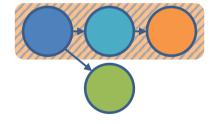
TP: Throughput

CP: Critical Path





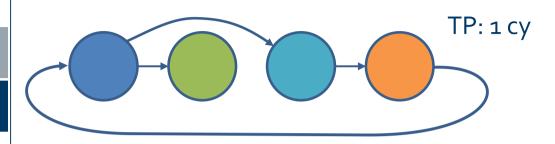
- Dependencies within loop
- Loop-carried dependencies



TP: Throughput

CP: Critical Path

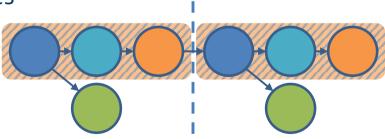




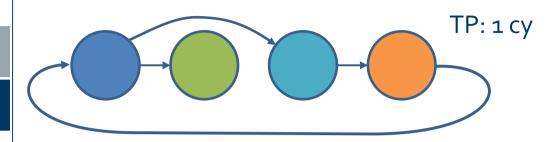
- Dependencies within loop
- Loop-carried dependencies

TP: Throughput

CP: Critical Path



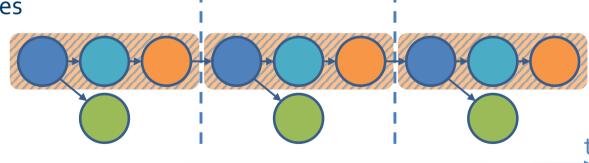




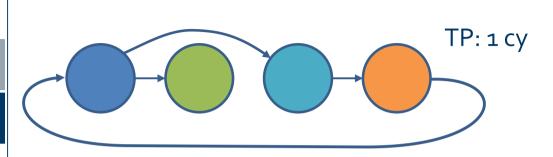
- Dependencies within loop
- Loop-carried dependencies

TP: Throughput

CP: Critical Path



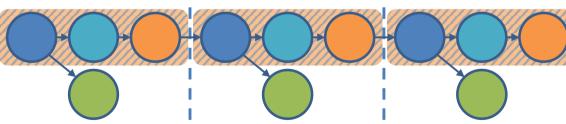






- Dependencies within loop
- Loop-carried dependencies

CP: 3 cy LCD: 3 cy

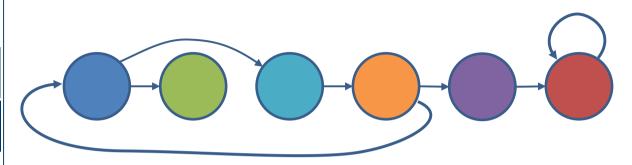


3 cy/it

TP: Throughput

CP: Critical Path



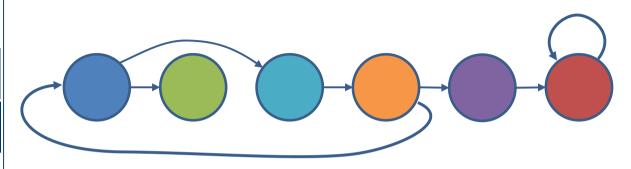


TP: 1 cy

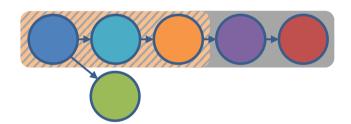
TP: Throughput

CP: Critical Path





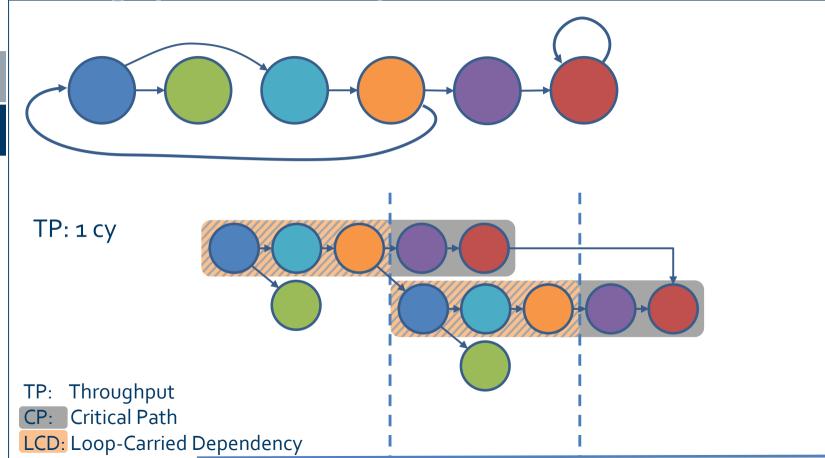
TP: 1 cy



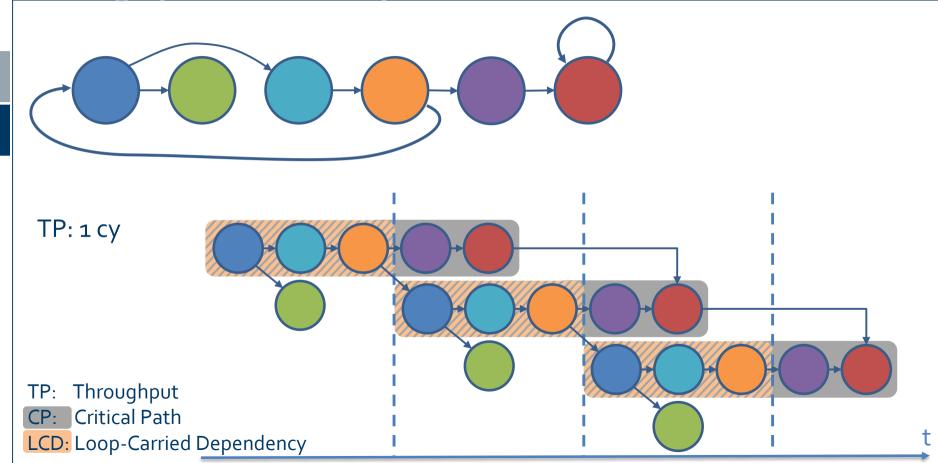
TP: Throughput

CP: Critical Path

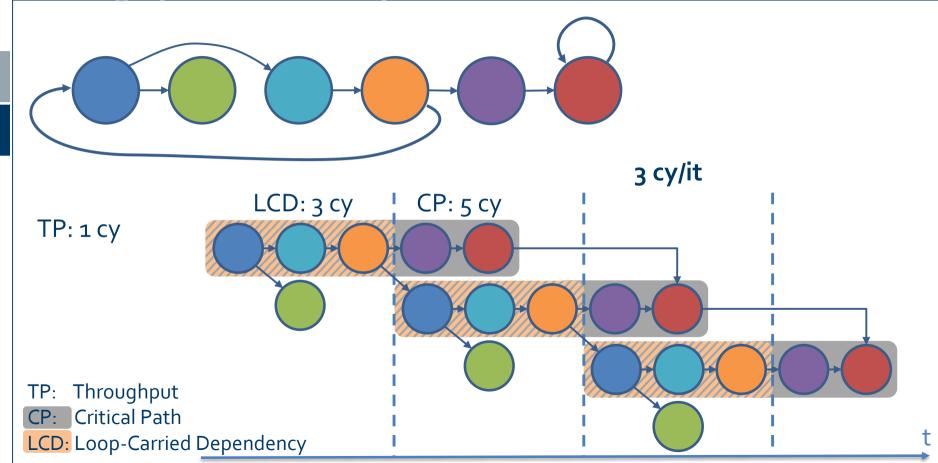














Marked Assembly

```
$111,%ebx
                           #START MARKER
      movl
      .byte
              100,103,144 #START MARKER
      .122:
        vmovapd 0(%r13,%rax),%ymm0
        vfmadd213pd (%r14,%rax),%ymm1,%ymm0
x86
        vmovapd %ymm0,(%r12,%rax)
        addq $32,%rax
        cmpq %rax, %r15
        jne .L22
      movl
              $222,%ebx
                           #END MARKER
      .bvte 100.103.144 #END MARKER
```

```
mov
             x1,#111 //START
      .byte 213,3,32,31 //START
       .L18:
        ldr q2, [x20, x0]
        ldr q1, [x21, x0]
arm
        fmla v1.2d, v2.2d, v0.2d
        str q1, [x19, x0]
        add x0, x0, #16
        cmp x22, x0
        bne .L18
             x1,#222
      mov
                          //END
       .byte 213,3,32,31 //END
```



Marked Assembly

```
$111.%ebx
                           #START MARKER
      mov1
      .byte
              100,103,144 #START MARKER
      .122:
        vmovapd 0(%r13,%rax),%ymm0
        vfmadd213pd (%r14,%rax),%ymm1,%ymm0
x86
        vmovapd %ymm0,(%r12,%rax)
        addq $32,%rax
        cmpq %rax, %r15
        jne .L22
              $222,%ebx
      movl
                           #END MARKER
      .bvte 100,103,144 #END MARKER
```

```
mov
             x1,#111
                     //START
      .byte 213,3,32,31 //START
       .L18:
        ldr q2, [x20, x0]
        ldr q1, [x21, x0]
arm
        fmla v1.2d, v2.2d, v0.2d
        str q1, [x19, x0]
        add x0, x0, #16
         cmp x22, x0
         bne .118
             x1,#222
      mov
                         //END
       .byte 213,3,32,31 //END
```

Intel Cascade Lake Out-of-Order Scheduler Port 0 Port 1 Port 5 Port 6 Port 2 Port 3 Port 4 Port 7 STORE SIMLPE AGU ALU ALU ALU ALU LOAD LOAD 2ND BRANCH FAST LEA FAST LEA 1ST BRANCH AGU AGU AVX FMA **AVX SHUF** AVX DIV AVX512 FMA AVX MUL AVX FMA AVX512 ADD AVX ADD AVX MUL AVX512 MUL AVX ADD AVX ALU AVX SHIFT AVX ALU AVX512 ALU **AVX SHIFT** Memory Control



Marked Assembly

```
$111.%ebx
                          #START MARKER
      movl
      .byte
             100,103,144 #START MARKER
      .122:
        vmovapd 0(%r13,%rax),%ymm0
        vfmadd213pd (%r14,%rax),%ymm1,%ymm0
x86
        vmovapd %ymm0,(%r12,%rax)
        addq $32,%rax
        cmpq %rax, %r15
        jne .L22
             $222,%ebx
      movl
                          #END MARKER
      .bvte 100,103,144 #END MARKER
       mov
             x1,#111 //START
       .byte 213,3,32,31 //START
       .L18:
        ldr q2, [x20, x0]
        ldr q1, [x21, x0]
arm
        fmla v1.2d, v2.2d, v0.2d
        str q1, [x19, x0]
        add x0, x0, #16
        cmp x22, x0
         bne .118
       mov
             x1,#222
                      //END
```

Machine Files / Databases

```
load latency: {gpr: 4, xmm: 4, ymm: 4, zmm: 4}
load throughput: {port pressure: [0,0,0,0.5 ... ,0]}
- name: vfmadd213pd
  operands:
  - class: "register"
   name: "ymm"
    source: true
   destination: false
  - class: "register"
   name: "ymm"
    source: true
   destination: false
  - class: "register"
   name: "ymm"
    source: true
    destination: true
throughput: 0.5
latency: 4 # 0 DV 1 2 D 3 D 4 5 6 7
port pressure: [0.5,0,0.5,0.5,0.5,0.5,0.5,0,0,0,0]
```

.byte 213,3,32,31 //END



Marked Assembly

```
$111.%ebx
                          #START MARKER
      movl
      .byte
             100,103,144 #START MARKER
      .122:
        vmovapd 0(%r13,%rax),%ymm0
        vfmadd213pd (%r14,%rax),%ymm1,%ymm0
x86
        vmovapd %ymm0,(%r12,%rax)
        addq $32,%rax
        cmpq %rax, %r15
        jne .L22
             $222,%ebx #END MARKER
      movl
      .bvte 100,103,144 #END MARKER
      mov
             x1.#111 //START
       .byte 213,3,32,31 //START
       .L18:
        ldr q2, [x20, x0]
        ldr q1, [x21, x0]
arm
        fmla v1.2d, v2.2d, v0.2d
        str q1, [x19, x0]
        add x0, x0, #16
        cmp x22, x0
         bne .118
             x1,#222
      mov
                      //END
       .byte 213,3,32,31 //END
```

Machine Files / Databases

```
load latency: {gpr: 4, xmm: 4, ymm: 4, zmm: 4}
load throughput: {port pressure: [0,0,0,0.5 ... ,0]}
- name: vfmadd213pd
  operands:
  - class: "register"
   name: "ymm"
   source: true
   destination: false
  - class: "register"
   name: "ymm"
   source: true
   destination: false
  - class: "register"
   name: "ymm"
    source: true
   destination: true
throughput: 0.5
latency: 4 # 0 DV 1 2 D 3 D 4 5 6 7
port pressure: [0.5,0,0.5,0.5,0.5,0.5,0.5,0,0,0,0]
```



Marked Assembly

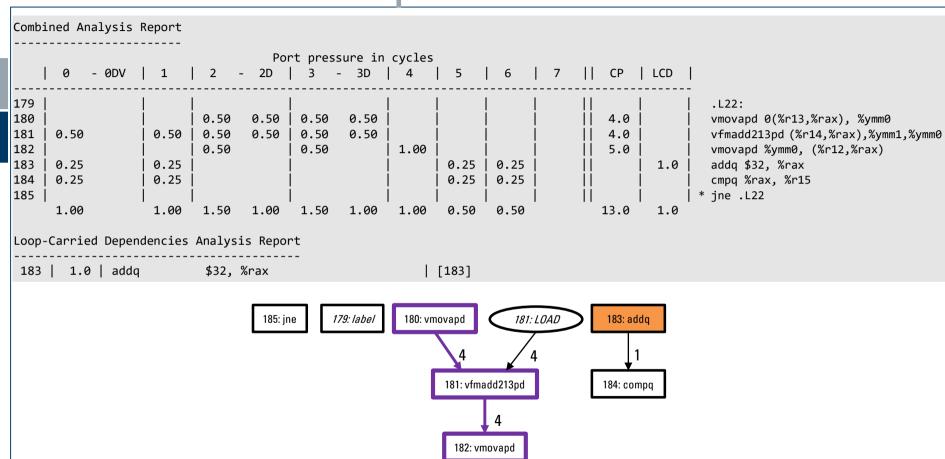
```
$111.%ebx
                       #START MARKER
      movl
      .byte
             100,103,144 #START MARKER
      .122:
        vmovapd 0(%r13,%rax),%ymm0
        vfmadd213pd (%r14,%rax),%ymm1,%ymm0
x86
        vmovapd %ymm0,(%r12,%rax)
        addq $32,%rax
        cmpq %rax, %r15
        jne .L22
             $222,%ebx #END MARKER
      movl
      .bvte 100,103,144 #END MARKER
      mov
             x1,#111 //START
       .byte 213,3,32,31 //START
       .L18:
        ldr q2, [x20, x0]
        ldr q1, [x21, x0]
arm
        fmla v1.2d, v2.2d, v0.2d
        str q1, [x19, x0]
        add x0, x0, #16
        cmp x22, x0
         bne .118
      mov
             x1,#222
                     //END
```

Machine Files / Databases

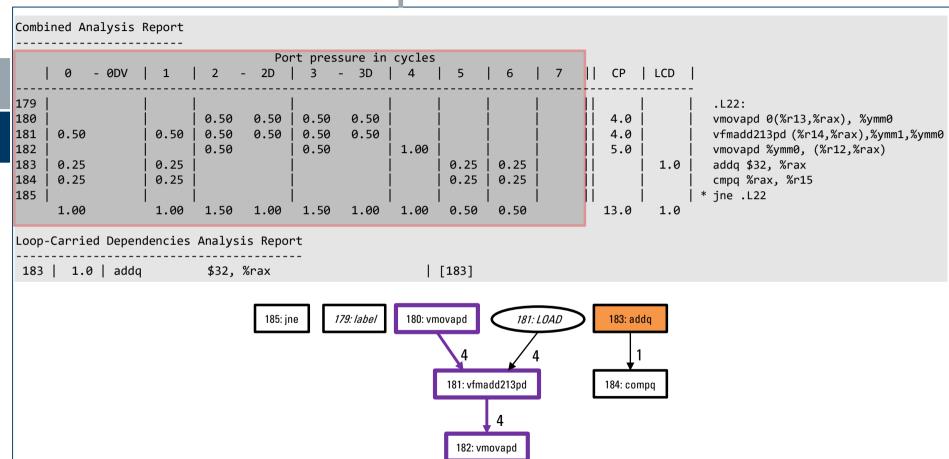
```
load latency: {gpr: 4, xmm: 4, ymm: 4, zmm: 4}
load throughput: {port pressure: [0,0,0,0.5 ... ,0]}
- name: vfmadd213pd
  operands:
  - class: "register"
   name: "ymm"
    source: true
   destination: false
  - class: "register"
   name: "ymm"
    source: true
   destination: false
  - class: "register"
   name: "ymm"
    source: true
   destination: true
throughput: 0.5
latency: 4 # 0 DV 1 2 D 3 D 4 5 6 7
port pressure: [0.5,0,0.5,0.5,0.5,0.5,0.5,0,0,0,0]
```

.byte 213,3,32,31 //END

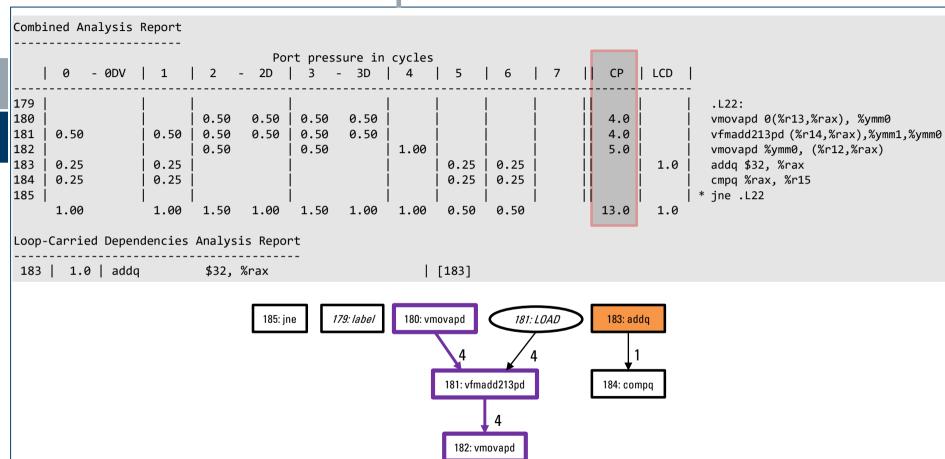




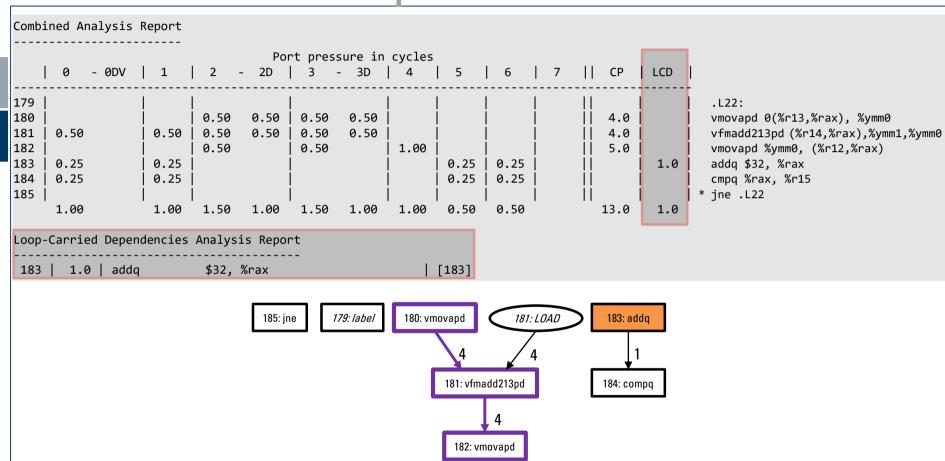




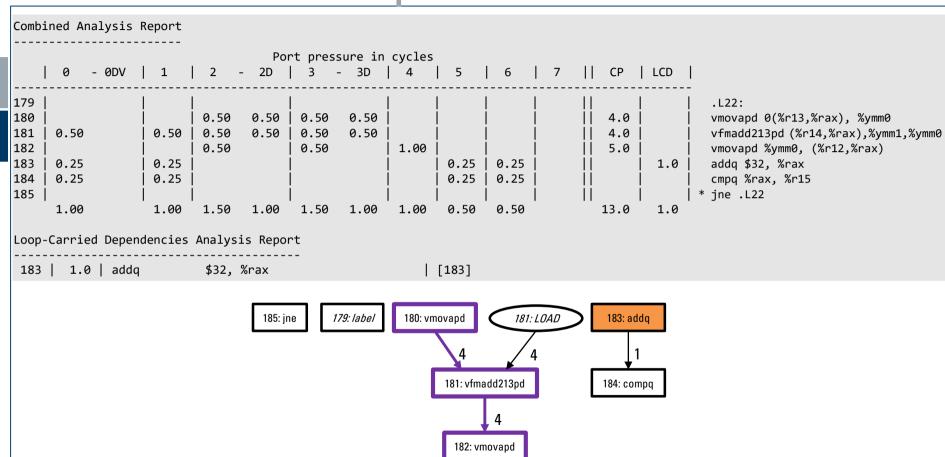






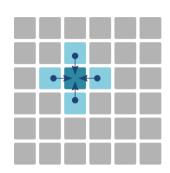








Limited by loop-carried dependency



 Create code with -Ofast, -funroll-loops (+ architecture specific flags)

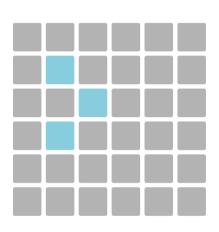
Analyze for Intel Cascake Lake,
 AMD Zen and
 Marvell ThunderX2

```
do it=1, itmax
  do k=1, kmax-1
     do i=1, imax-1
     phi(i,k,t0) = 0.25 * (
        phi(i,k-1,t0) + phi(i+1,k,t0) +
        phi(i,k+1,t0) + phi(i-1,k,t0))
     do
  do
  do
```



```
x1, #111
                   // START MARKER
 mov
  .byte 213,3,32,31 // START MARKER
.L20:
 ldr
         d31, [x15, x18, lsl 3]
 ldr
         d0, [x15, 8]
         x14, x15
 mov
         x16, x15, 24
 add
 ldr
         d2, [x15, x30, lsl 3]
  add
         x15, x15, 32
 fadd
         d1, d31, d0
 fadd
         d3, d1, d30
 fadd
         d4, d3, d2
         d5, d4, d9
 fmul
  str
         d5, [x14], 8
         d6, [x14, x18, lsl 3]
 ldr
 ldr
         d16, [x14, 8]
         x13, x14, 8
 add
 ldr
         d7, [x14, x30, lsl 3]
 fadd
         d17, d6, d16
         d18, d17, d5
 fadd
         d19, d18, d7
 fadd
         d20, d19, d9
 fmul
         d20, [x15, -24]
  str
```

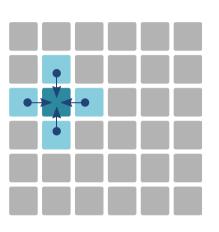
```
ldr
       d21, [x13, x18, lsl 3]
ldr
       d23, [x14, 16]
       d22, [x13, x30, lsl 3]
ldr
       d24, d21, d23
fadd
       d25, d24, d20
fadd
fadd
       d26, d25, d22
fmul
       d27, d26, d9
str
       d27, [x14, 8]
       d30, [x15]
ldr
ldr
       d28, [x16, x18, lsl 3]
ldr
       d29, [x16, x30, lsl 3]
       d31, d28, d30
fadd
       d2, d31, d27
fadd
       d0, d2, d29
fadd
       d30, d0, d9
fmul
str
       d30, [x15, -8]
       x7, x15
cmp
       .L20
bne
      x1, #222 // END MARKER
mov
.byte 213,3,32,31 // END MARKER
```





```
x1, #111
                   // START MARKER
 mov
  .byte 213,3,32,31 // START MARKER
.L20:
 ldr
         d31, [x15, x18, lsl 3]
 ldr
         d0, [x15, 8]
         x14, x15
 mov
         x16, x15, 24
 add
 ldr
         d2, [x15, x30, lsl 3]
  add
         x15, x15, 32
 fadd
         d1, d31, d0
 fadd
         d3, d1, d30
 fadd
         d4, d3, d2
         d5, d4, d9
 fmul
  str
         d5, [x14], 8
         d6, [x14, x18, lsl 3]
 ldr
 ldr
         d16, [x14, 8]
         x13, x14, 8
 add
 ldr
         d7, [x14, x30, lsl 3]
 fadd
         d17, d6, d16
         d18, d17, d5
 fadd
         d19, d18, d7
 fadd
         d20, d19, d9
 fmul
         d20, [x15, -24]
  str
```

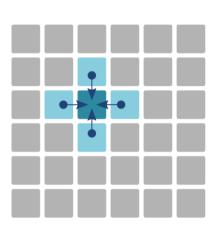
```
ldr
       d21, [x13, x18, lsl 3]
ldr
       d23, [x14, 16]
       d22, [x13, x30, lsl 3]
ldr
       d24, d21, d23
fadd
fadd
       d25, d24, d20
       d26, d25, d22
fadd
fmul
       d27, d26, d9
str
       d27, [x14, 8]
       d30, [x15]
ldr
ldr
       d28, [x16, x18, lsl 3]
ldr
       d29, [x16, x30, lsl 3]
       d31, d28, d30
fadd
       d2, d31, d27
fadd
       d0, d2, d29
fadd
       d30, d0, d9
fmul
str
       d30, [x15, -8]
       x7, x15
cmp
       .L20
bne
      x1, #222
               // END MARKER
mov
.byte 213,3,32,31 // END MARKER
```





```
x1, #111
                    // START MARKER
 mov
  .byte 213,3,32,31 // START MARKER
.L20:
 ldr
         d31, [x15, x18, lsl 3]
 ldr
         d0, [x15, 8]
         x14, x15
 mov
        x16, x15, 24
 add
 ldr
         d2, [x15, x30, lsl 3]
         x15, x15, 32
  add
 fadd
         d1, d31, d0
 fadd
         d3, d1, d30
 fadd
         d4, d3, d2
         d5, d4, d9
 fmul
  str
         d5, [x14], 8
         d6, [x14, x18, lsl 3]
 ldr
         d16, [x14, 8]
 ldr
         x13, x14, 8
 add
 ldr
         d7, [x14, x30, lsl 3]
 fadd
         d17, d6, d16
         d18, d17, d5
 fadd
         d19, d18, d7
 fadd
         d20, d19, d9
 fmul
         d20, [x15, -24]
  str
```

```
ldr
       d21, [x13, x18, lsl 3]
ldr
       d23, [x14, 16]
       d22, [x13, x30, lsl 3]
ldr
       d24, d21, d23
fadd
       d25, d24, d20
fadd
       d26, d25, d22
fadd
fmul
       d27, d26, d9
str
       d27, [x14, 8]
       d30, [x15]
ldr
ldr
       d28, [x16, x18, lsl 3]
ldr
       d29, [x16, x30, lsl 3]
       d31, d28, d30
fadd
       d2, d31, d27
fadd
       d0, d2, d29
fadd
       d30, d0, d9
fmul
str
       d30, [x15, -8]
       x7, x15
cmp
       .L20
bne
      x1, #222
               // END MARKER
mov
.byte 213,3,32,31 // END MARKER
```

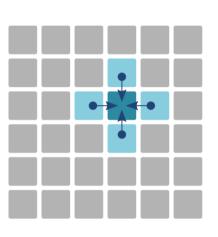


Gauss-Seidel Method Example



```
x1, #111
                     // START MARKER
 mov
  .byte 213,3,32,31 // START MARKER
.L20:
 ldr
         d31, [x15, x18, lsl 3]
 ldr
         d0, [x15, 8]
         x14, x15
 mov
        x16, x15, 24
 add
 ldr
         d2, [x15, x30, lsl 3]
  add
         x15, x15, 32
 fadd
         d1, d31, d0
 fadd
         d3, d1, d30
 fadd
         d4, d3, d2
         d5, d4, d9
 fmul
  str
         d5, [x14], 8
         d6, [x14, x18, lsl 3]
 ldr
 ldr
         d16, [x14, 8]
 add
         x13, x14, 8
 ldr
         d7, [x14, x30, lsl 3]
 fadd
         d17, d6, d16
         d18, d17, d5
 fadd
         d19, d18, d7
 fadd
         d20, d19, d9
 fmul
         d20, [x15, -24]
  str
```

```
ldr
       d21, [x13, x18, lsl 3]
ldr
       d23, [x14, 16]
       d22, [x13, x30, lsl 3]
ldr
       d24, d21, d23
fadd
       d25, d24, d20
fadd
fadd
       d26, d25, d22
fmul
       d27, d26, d9
       d27, [x14, 8]
str
ldr
       d30, [x15]
ldr
       d28, [x16, x18, lsl 3]
ldr
       d29, [x16, x30, lsl 3]
       d31, d28, d30
fadd
       d2, d31, d27
fadd
       d0, d2, d29
fadd
       d30, d0, d9
fmul
str
       d30, [x15, -8]
       x7, x15
cmp
       .L20
bne
      x1, #222
                    // END MARKER
mov
.byte 213,3,32,31 // END MARKER
```

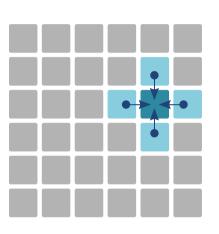


Gauss-Seidel Method Example



```
x1, #111
                   // START MARKER
 mov
  .byte 213,3,32,31 // START MARKER
.L20:
 ldr
         d31, [x15, x18, lsl 3]
 ldr
         d0, [x15, 8]
         x14, x15
 mov
        x16, x15, 24
 add
 ldr
         d2, [x15, x30, lsl 3]
  add
         x15, x15, 32
 fadd
         d1, d31, d0
 fadd
         d3, d1, d30
 fadd
         d4, d3, d2
 fmul
         d5, d4, d9
  str
         d5, [x14], 8
         d6, [x14, x18, lsl 3]
 ldr
 ldr
         d16, [x14, 8]
 add
         x13, x14, 8
 ldr
         d7, [x14, x30, lsl 3]
 fadd
         d17, d6, d16
 fadd
         d18, d17, d5
         d19, d18, d7
 fadd
         d20, d19, d9
 fmul
         d20, [x15, -24]
  str
```

```
ldr
       d21, [x13, x18, lsl 3]
ldr
       d23, [x14, 16]
       d22, [x13, x30, lsl 3]
ldr
       d24, d21, d23
fadd
fadd
       d25, d24, d20
fadd
       d26, d25, d22
fmul
       d27, d26, d9
str
       d27, [x14, 8]
ldr
       d30, [x15]
ldr
       d28, [x16, x18, lsl 3]
ldr
       d29, [x16, x30, lsl 3]
       d31, d28, d30
fadd
       d2, d31, d27
fadd
fadd
       d0, d2, d29
       d30, d0, d9
fmul
       d30, [x15, -8]
str
       x7, x15
cmp
       .L20
bne
      x1, #222
               // END MARKER
mov
.byte 213,3,32,31 // END MARKER
```









	0 - 0DV	Port p 1 - 1DV	ressure 2	in cyc.	les 4	5	CP	LCD	l	
520 521 522 523 524 525	0.50 0.33	0.50 0.33	0.33	0.50 0.50 0.50	0.50 0.50 0.50		4.0		L20: ldr ldr mov add ldr	d31, [x15, x18, 1s1 3] d0, [x15, 8] x14, x15 x16, x15, 24 d2, [x15, x30, 1s1 3]
5224 5224 55226 55227 5527 55227 55227 55227 55227 55227 55227 55227 55227 55227 55227 5527 5527 55227 55227 55227 55227 55227 55227 55227 55227 55227 55227 5527 55227 55227 55227 55227 55227 55227 55227 55227 55227 55227 5527	0.33 0.50 0.50 0.50 0.50	0.33 0.50 0.50 0.50 0.50	0.33	0.50	0.50	1.00	6.0 6.0 6.0 4.0	6.0 6.0 6.0	add fadd fadd fadd fmul str	x15, x15, 32 d1, d31, d0 d3, d1, d30 d4, d3, d2 d5, d4, d9 d5, [x14], 8
532 533 534	0.33	0.33	0.33	0.50 0.50 0.50	0.50 0.50 0.50		4.0		ldr ldr add ldr	d6, [x14, x18, lsl 3] d16, [x14, 8] x13, x14, 8
536 537 538 539	0.50 0.50 0.50 0.50	0.50 0.50 0.50 0.50				1 00	6.0 6.0 6.0	6.0 6.0 6.0	fadd fadd fadd fmul	d7, [x14, x30, ls1 3] d17, d6, d16 d18, d17, d5 d19, d18, d7 d20, d19, d9
540 541 542 543 544	0.50	0.50		0.50 0.50 0.50 0.50	0.50 0.50 0.50 0.50	1.00			str ldr ldr ldr fadd	d20, [x15, -24] d21, [x13, x18, lsl 3] d23, [x14, 16] d22, [x13, x30, lsl 3] d24, d21, d23
545 546 547 548	0.50 0.50 0.50	0.50 0.50 0.50		0.50	0.50	1.00	6.0 6.0 6.0	6.0 6.0 6.0	fadd fadd fmul str	d25, d24, d20 d26, d25, d22 d27, d26, d9 d27, [x14, 8]
549 550 551 552	0.50	0.50		0.50 0.50 0.50	0.50 0.50 0.50				ldr ldr ldr fadd	d30, [x15] d28, [x16, x18, lsl 3] d29, [x16, x30, lsl 3] d31, d28, d30
553 554 555 556	0.50 0.50 0.50	0.50 0.50 0.50		0.50	0.50	1.00	6.0 6.0 6.0 4.0	6.0 6.0 6.0	fadd fadd fmul str	d2, d31, d27 d0, d2, d29 d30, d0, d9 d30, [x15, -8]
557 558	0.33	0.33	0.33						* bne	x7, x15 .L20
Loop-	9.83 Carried Depend	9.83 dencies Analys:	1.33 is Repo	8.00 rt	8.00	4.00	100.0	72.0		
526 555	1.0 add 72.0 fmul	x15, x15, 32 d30, d0, d9				[526] [528, 5	529, 530,	537, !	538, 539,	545, 546, 547, 553, 554, 555]







	0 - 0	Port DDV 1 - 1DV	pressure 2	in cyc	les 4	5	CP	LCD	I		
520 521 522	0.50	0.50		0.50	0.50		4.0		L20:	d31, [x15, x18, lsl 3] d0, [x15, 8] x14, x15	
522 523 524 525 526 527 528 529	0.33	0.33	0.33	0.50	0.50		: :		mov add ldr	x16, x15, 24	
526 527	0.33 0.50	0.33 0.50	0.33	0.50	0.50		6.0		l add	x15, x15, 32 d1, d31, d0	
528 529	0.50 0.50 0.50	0.50 0.50					6.0	6.0	fadd fadd fadd	d3, d1, d30 d4, d3, d2	
530 531 532 533	0.50	0.50		0.50 0.50 0.50	0.50 0.50 0.50	1.00	6.0 4.0 4.0	6.0	fmul str ldr ldr	d2, [x15, x30, lsl 3] x15, x15, 32 d1, d31, d0 d3, d1, d30 d4, d3, d2 d5, d4, d9 d5, [x14], 8 d6, [x14, x18, lsl 3] d16, [x14, 8] x13, x14, 8 d7, [x14, x30, lsl 3] d17, d6, d16 d18, d17, d5 d19, d18, d7 d20, d19, d9	
534 535	0.33	0.33	0.33	0.50	0.50				add ldr	x13, x14, 8 d7, [x14, x30, lsl 3]	
533 534 535 536 537 538 539	0.50 0.50 0.50 0.50	0.50 0.50 0.50 0.50					6.0 6.0 6.0 6.0	6.0 6.0	fadd fadd fadd fmul	d17, d6, d16 d18, d17, d5 d19, d18, d7 d20, d19, d9 d20, [x15, -24]	
540 541 542 543				0.50 0.50 0.50 0.50	0.50 0.50 0.50 0.50	1.00			str ldr ldr ldr	d21, [x13, x18, lsl 3] d23, [x14, 16]	
543 544 545 546	0.50 0.50	0.50 0.50					6.0 6.0	6.0	l fadd	d22, [x13, x30, ls1 3] d24, d21, d23 d25, d24, d20 d26, d25, d22	
546 547	0.50 0.50	0.50 0.50		0 50	0.50	1.00	6.0 6.0	6.0	fadd fadd fmul str	d26, d25, d22 d27, d26, d9 d27, [x14, 8]	
547 548 549 550				0.50 0.50 0.50	0.50	1.00]]		l ldr ldr	d30, [X15] d28 [x16 x18]s1 31	DI I
551 552 553 554 555 556 557 558	0.50 0.50 0.50	0.50 0.50		0.50	0.50		6.0	6.0	ldr fadd fadd fadd	d29, [x16, x30, 151 3] d31, d28, d30 d2, d31, d27 d0, d2, d29 d30, d0, d9	Block
554 555	0.50	0.50 0.50					[] 6.0	6.0 6.0	fmul	d0, d2, d29 d30, d0, d9	
556 557 558	0.33	0.33	0.33	0.50	0.50	1.00	4.0		str cmp * bne	d30, [x15, -8] x7, x15 .L20	
<	9.83	9.83	1.33	8.00	8.00	4.00	100.0	72.0			
Loop-	-Carried D	ependencies Anal	ysis Repo	rt							
526 555	1.0 a	add x15, x15, mul d30, d0,	32 d9			[526] [528]	529, 530.	537, 5	538, 539,	545, 546, 547, 553, 554,	555]

k Throughput

18.11.2019

2.46 cy

PMBS19 | OSACA | Jan Laukemann







====	0 - 0DV	Port 1 - 1D\	pressure / 2	in cyc 3 	les 4 	5	CP	LCD	<u> </u>		
520 521 522 523 524 525 526 527 528	0.50 0.33	0.50 0.33	0.33	0.50 0.50 0.50	0.50 0.50 0.50		4.0		l .L20: ldr ldr mov add ldr	d31, [x15, x18, lsl 3] d0, [x15, 8] x14, x15 x16, x15, 24 d2, [x15, x30, lsl 3]	
526 527 528 529 530 531 532	0.33 0.50 0.50 0.50 0.50	0.33 0.50 0.50 0.50 0.50	0.33	0.50 0.50	0.50 0.50	1.00	6.0 6.0 6.0 6.0 4.0 4.0	6.0 6.0 6.0	add fadd fadd fadd fmul str ldr	x15, x15, 32 d1, d31, d0 d3, d1, d30 d4, d3, d2 d5, d4, d9 d5, [x14], 8 d6, [x14, x18, lsl 3]	
529 530 531 532 533 534 535 536 537 538 539 540	0.33 0.50 0.50 0.50	0.33 0.50 0.50 0.50	0.33	0.50	0.50		6.0 6.0 6.0	6.0 6.0	ldr add ldr fadd fadd fadd	d16, [x14, 8] x13, x14, 8 d7, [x14, x30, 1s1 3] d17, d6, d16 d18, d17, d5 d19, d18, d7	
541 542 543	0.50	0.50		0.50 0.50 0.50 0.50	0.50 0.50 0.50 0.50	1.00	6.0	6.0	fmul str ldr ldr ldr fadd	d20, d19, d9 d20, [x15, -24] d21, [x13, x18, lsl 3] d23, [x14, 16] d22, [x13, x30, lsl 3]	
544 545 546 547 548 549 550	0.50 0.50 0.50 0.50	0.50 0.50 0.50 0.50		0.50 0.50	0.50 0.50	1.00	6.0 6.0 6.0	6.0 6.0 6.0	fadd fadd fmul str ldr	d24, d21, d23 d25, d24, d20 d26, d25, d22 d27, d26, d9 d27, [x14, 8] d30, [x15]	
550 551 552 553 554 555	0.50 0.50 0.50	0.50 0.50 0.50		0.50	0.50		6.0	6.0	ldr ldr fadd fadd fadd	d28, [x16, x18, lsl 3] d29, [x16, x30, lsl 3] d31, d28, d30 d2, d31, d27 d0, d2, d29	Block
555 556 557 558	0.50 0.33	0.50	0.33	0.50	0.50	1.00	6.0	6.0	fmul str cmp * bne	d30, d0, d9 d30, [x15, -8] x7, x15 .L20	Critica
Loop-	9.83 -Carried Depe	9.83 endencies Anal	1.33 Lysis Repo	8.00 rt	8.00	4.00	100	72.0			
526 555	1.0 add 72.0 fmu]	x15, x15, l d30, d0,	32 d9		I	[526] [528, 5	29, 530	, 537, !	538, 539,	545, 546, 547, 553, 554, 555]	

k Throughput 2.46 cy al Path 25.0 cy

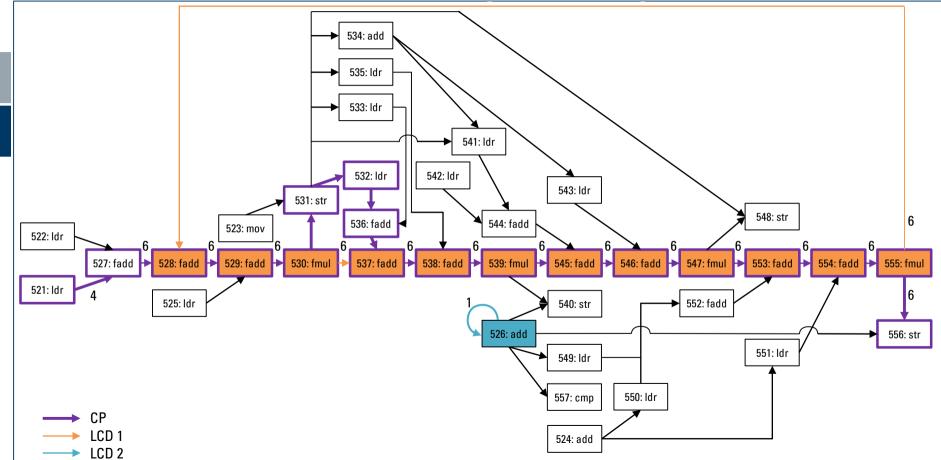






	0 - 0DV	Port p 1 - 1DV	ressure 2	in cycl	les 4	5	CP	LCD						
520 521 522	a 5a	0.50		0.50 0.50	0.50 0.50		4.0		.L20: ldr ldr	d31, [x15, x18 d0, [x15, 8]	, lsl 3]		OS.	ACA
524 525	0.50 0.33	0.33	0.33	0.50	0.50				mov add ldr	d0, [x15, 8] x14, x15 x16, x15, 24 d2, [x15, x30, x15, x15, 32	lsl 3]		_	
526 527	0.33 0.50	0.33 0.50 0.50 0.50	0.33				6.0		add fadd	x15, x15, 32 d1, d31, d0 d3, d1, d30	•			
528 529 530	0.33 0.50 0.50 0.50 0.50	0.50 0.50 0.50					6.0 6.0 6.0 4.0	6.0 6.0 6.0	fadd fadd fmul str	d1, d31, d0 d3, d1, d30 d4, d3, d2 d5, d4, d9 d5, [x14], 8				
531 532				0.50	0.50	1.00	4.0		ldr	d5, [x14], 8 d6, [x14, x18,	lsl 3]			
533 534 535	0.33	0.33	0.33	0.50	0.50 0.50				ldr add ldr	d6, [x14, x18, d16, [x14, 8] x13, x14, 8 d7, [x14, x30,	1s1 31			
520 5212 5223 5225 5226 5228 5229 5331 5332 5334 5335 5337 5349 5443 5445 5447 5449 5553 5555 5556 5558	0.50 0.50 0.50 0.50	0.50 0.50 0.50					6.0 6.0 6.0	6.0	fadd fadd fadd fmul	d17, d6, d16 d18, d17, d5 d19, d18, d7	-5- 0,			
539 540 541	0.50	0.50		0.50 0.50 0.50	0.50 0.50 0.50 0.50	1.00	6.0	6.0	str ldr	d20, d19, d9 d20, [x15, -24 d21, [x13, x18] , lsl 3]			
542 543	0.50	0.50		0.50 0.50	0.50 0.50				ldr ldr fadd	d23, [x14, 16] d22, [x13, x30 d24, d21, d23				
545 546	0.50 0.50	0.50 0.50					6.0	6.0 6.0	fadd fadd fadd	d25, d24, d20 d26, d25, d22				
547 548	0.50	0.50		0.50 0.50 0.50	0.50 0.50 0.50	1.00	6.0	6.0	fmul str	d27, d26, d9 d27, [x14, 8]				
550 551				0.50	0.50				fadd fadd fmul str ldr ldr fadd	d28. [x16] x18	, lsl 3] , lsl 3]	Block Thr	roughput	2.46 cy
552 553	0.50 0.50 0.50 0.50	0.50 0.50 0.50 0.50					6.0	6.0 6.0	fadd fadd fadd fmul	d29, [x16, x30 d31, d28, d30 d2, d31, d27 d0, d2, d29	•			
555 556	0.50	0.50		0.50	0.50	1.00	6.0	6.0	fmul str	d30, d0, d9 d30, [x15, -8]		Critical P	ath	25.0 cy
557 558	0.33	0.33	0.33						cmp * bne	x7, x15 .L20		Loop Cor	ried Den	10.0 00
	9.83	9.83	1.33	8.00	8.00	4.00	100	72)			Loop-Car	ried Dep.	18.0 cy
	-Carried Depend			rt 										
526 555	1.0 add 72.0 fmul	x15, x15, 32 d30, d0, d9				[526] [528, 5	29, 530,	537, 5	38, 539,	545, 546, 547,	553, 554, 555]			







		Measu	Prediction [cy/it]										
Architecture	Unroll factor	ivieasu			OSACA			IACA			LLVM-M		
		MLUP/s	cy/it	TP	LCD	CP	TP	LCD	СР	TP	LCD	CP	
Intel Cascade Lake X	4x	178.3	14.02	2.19	14.0	18.0	2.0 (14.0)		2.0	14.75	19.0	
AMD Zen	4x	194.4	11.83		11.5					3.0	18.0	24.0	
Marvell ThunderX2	4x	118.9	18.50	2.46	18.0	25.0							



		Magau	Prediction [cy/it]									
Architecture	Unroll factor	Measured		OSACA			IACA			LL\	CA	
		MLUP/s	cy/it	TP	LCD	СР	TP	LCD	СР	TP	LCD	СР
Intel Cascade Lake X	4x	178.3	14.02	2.19	14.0	18.0	2.0 (14.0)		2.0	14.75	19.0
AMD Zen	4x	194.4	11.83	2.0	11.5	15.0				3.0	18.0	24.0
Marvell ThunderX2	4x	118.9	18.50	2.46	18.0	25.0						



		Magau	Prediction [cy/it]									
Architecture	Unroll factor	Measured		OSACA			IACA			LL\	CA	
		MLUP/s	cy/it	TP	LCD	CP	TP	LCD	СР	TP	LCD	СР
Intel Cascade Lake X	4x	178.3	14.02	2.19	14.0	18.0	2.0 (14.0)		2.0	14.75	19.0
AMD Zen	4x	194.4	11.83	2.0	11.5	15.0				3.0	18.0	24.0
Marvell ThunderX2	4x	118.9	18.50	2.46	18.0	25.0						



		Magau		Prediction [cy/it]								
Architecture	Unroll factor	Measured		OSACA			IACA			LL\	CA	
		MLUP/s	cy/it	TP	LCD	СР	TP	LCD	СР	TP	LCD	CP
Intel Cascade Lake X	4x	178.3	14.02	2.19	14.0	18.0	2.0 (14.0)		2.0	14.75	19.0
AMD Zen	4x	194.4	11.83	2.0	11.5	15.0				3.0	18.0	24.0
Marvell ThunderX2	4x	118.9	18.50	2.46	18.0	25.0						

Summary – OSACA



- Automatic extraction, throughput and critical path analysis
- Cross-platform (Intel, AMD, ARM)
- Accurate predictions
- Open Source
- Allows architectural exploration

Future Work



- Support of hidden dependencies
- More precise LCD analysis
- Support new micro-architectures (Zen 2, Power 9, ...)
- More precise latency analysis for FMA instructions
- Considering ROB, register renaming, retirement, ...
- Optimally balanced port utilization

Future Work



- Support of hidden dependencies
- More precise LCD analysis
- Support new micro-architectures (Zen 2, Power 9, ...)
- More precise latency analysis for FMA instructions
- Considering ROB, register renaming, retirement, ...
- Optimally balanced port utilization



Future Work



- Support of hidden dependencies
- More precise LCD analysis
- Support new micro-architectures (Zen 2, Power 9, ...)
- More precise latency analysis for FMA instructions
- Considering ROB, register renaming, retirement, ...
- Optimally balanced port utilization





github.com/RRZE-HPC/OSACA



Reproduce at: https://github.com/RRZE-HPC/OSACA-CP-2019/