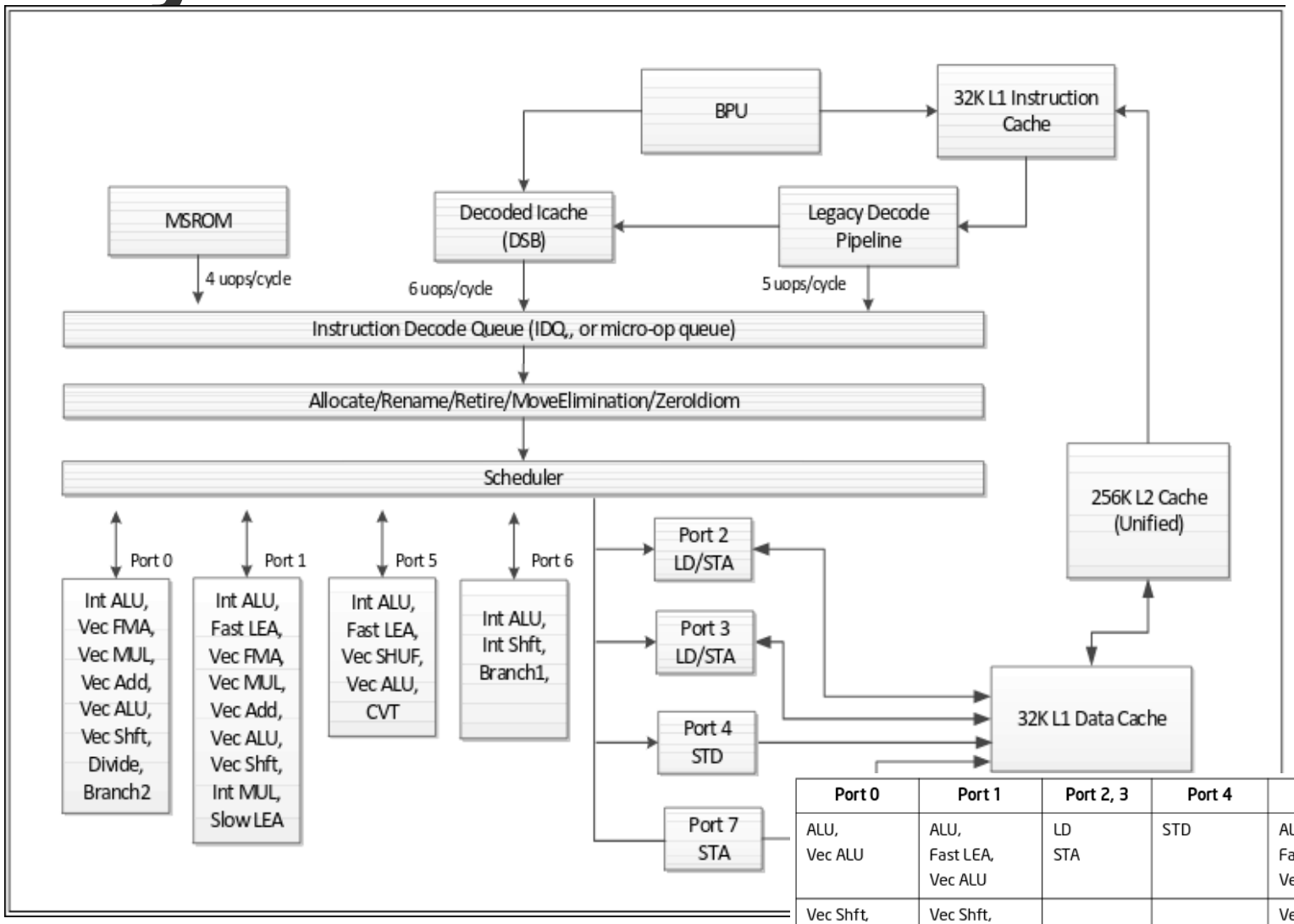
X86 resursmodell

Skylake microarchitecture

Execution Unit	# of Unit	Instructions
ALU	4	add, and, cmp, or, test, xor, movzx, movsx, mov, (v)movdqu, (v)movdqa, (v)movap*, (v)movup*
SHFT	2	sal, shl, rol, adc, sarx, adcx, adox, etc.
Slow Int	1	mul, imul, bsr, rcl, shld, mulx, pdep, etc.
BM	2	andn, bextr, blsi, blsmsk, bzhi, etc
Vec ALU	3	(v)pand, (v)por, (v)pxor, (v)movq, (v)movq, (v)movap*, (v)movup*, (v)andp*, (v)orp*, (v)paddb/w/d/q, (v)blendv*, (v)blendp*, (v)pblendd
Vec_Shft	2	(v)psllv*, (v)psrlv*, vector shift count in imm8
Vec Add	2	(v)addp*, (v)cmpp*, (v)max*, (v)min*, (v)padds*, (v)paddus*, (v)psign, (v)pabs, (v)pavgb, (v)pcmpeq*, (v)pmax, (v)cvtps2dq, (v)cvtdq2ps, (v)cvtsd2si, (v)cvtss2si
Shuffle	1	(v)shufp*, vperm*, (v)pack*, (v)unpck*, (v)punpck*, (v)pshuf*, (v)pslldq, (v)alignr, (v)pmovzx*, vbroadcast*, (v)pslldq, (v)psrldq, (v)pblendw
Vec Mul	2	(v)mul*, (v)pmul*, (v)pmadd*,
SIMD Misc	1	STTNI, (v)pclmulqdq, (v)psadw, vector shift count in xmm,
FP Mov	1	(v)movsd/ss, (v)movd gpr,
DIVIDE	1	divp*, divs*, vdiv*, sqrt*, vsqrt*, rcp*, vrcp*, rsqrt*, idiv

Skylake microarchitecture



Port 0	Port 1	Port 2, 3	Port 4	Port 5	Port 6	Port 7
ALU, Vec ALU	ALU, Fast LEA, Vec ALU	LD STA	STD	ALU, Fast LEA, Vec ALU,	ALU, Shft,	STA
Vec Shft, Vec Add,	Vec Shft, Vec Add,			Vec Shuffle,	Branch1	
Vec Mul, FMA,	Vec Mul, FMA					
DIV,	Slow Int					
Branch2	Slow LEA					

Agner Fog resursmodell

Integer instructions

Instruction	Operands	µops fused domain	µops unfused domain	µops each port	Latency	Reciprocal throughput	Comments
Move instructions							
MOV	r,i	1	1	p0156		0.25	
MOV	r8/16,r8/16	1	1	p0156	1	0.25	
MOV	r32/64,r32/64	1	1	p0156	0-1	0.25	may be elim.
MOV	r8l,m	1	2	p23 p0156		0.5	
MOV	r8h,m	1	1	p23		0.5	
MOV	r16,m	1	2	p23 p0156		0.5	
MOV	r32/64,m	1	1	p23	2	0.5	all addressing modes
MOV	m,r	1	2	p237 p4	2	1	
MOV	m,i	1	2	p237 p4		1	

Instruction name

Physical resources used

Latency

Throughput, <1?

Agner Fog resursmodell nackdelar

- Enbart troughput på instruktionsnivå (inte för micro-ops).
- Tabeller svåra att parse’a (specialtecken etc.)
- Få instruktioner inkluderade
- Ofullständiga rader/instruktioner
- Namn inte ‘mappade’ till LLVM-namn

LLVM 6.0.0 resursmodell

“Target” nivå (X86)

```
def MUL16r : I<0xF7, MRM4r, (outs),  (ins GR16:$src),
                 "mul{w}\t$src",
                 [], IIC_MUL16_REG>, OpSize16, Sched<[WriteIMul]>;
```

X86InstrArithmetic.td

```
defm : SKLWriteResPair<WriteIMul,  SKLPort1,   3>; // Integer multiplication.
```

X86SchedSkylakeClient.td

“Sub-target” nivå (skylake)

```
def SKLWriteResGroup112 : SchedWriteRes<[SKLPort0,SKLPort5,SKLPort23]> {
  let Latency = 8;
  let NumMicroOps = 4;
  let ResourceCycles = [1,2,1];
}
def: InstRW<[SKLWriteResGroup112], (instregex "MMX_PHADDSWrm64")>;
def: InstRW<[SKLWriteResGroup112], (instregex "MMX_PHSUBSWrm64")>;
```

X86SchedSkylakeClient.td

Manuella definitioner

```
{
        Instruction: CMOV_FR32,
        ResourceGroup: SKLWriteResGroup7
},
{
        Instruction: CMOV_GR8,
        ResourceGroup: SKLWriteResGroup7
},
{
        Instruction: COPY,
        ResourceGroup: WriteMove
},
```

manual_instruction_mapping.json

LLVM resursmodell nackdelar

- Förutsätter att alla micro-operationer har en throughput av **1** vid användandet av en resurs.
- Mikro-operationer kan vara beroende av resultat från mikro-operationer från samma instruktion.
- Fortfarande många instruktioner som saknas i resursmodellen.
- Kan inte garantera vilken resurs utav flera möjliga som används.

Retrospekt

- Spendera mer tid med ‘förstudie’, första sättet är inte alltid bästa
  - Agner vs LLVM
  - Parse’a fil vs. använda tablegen