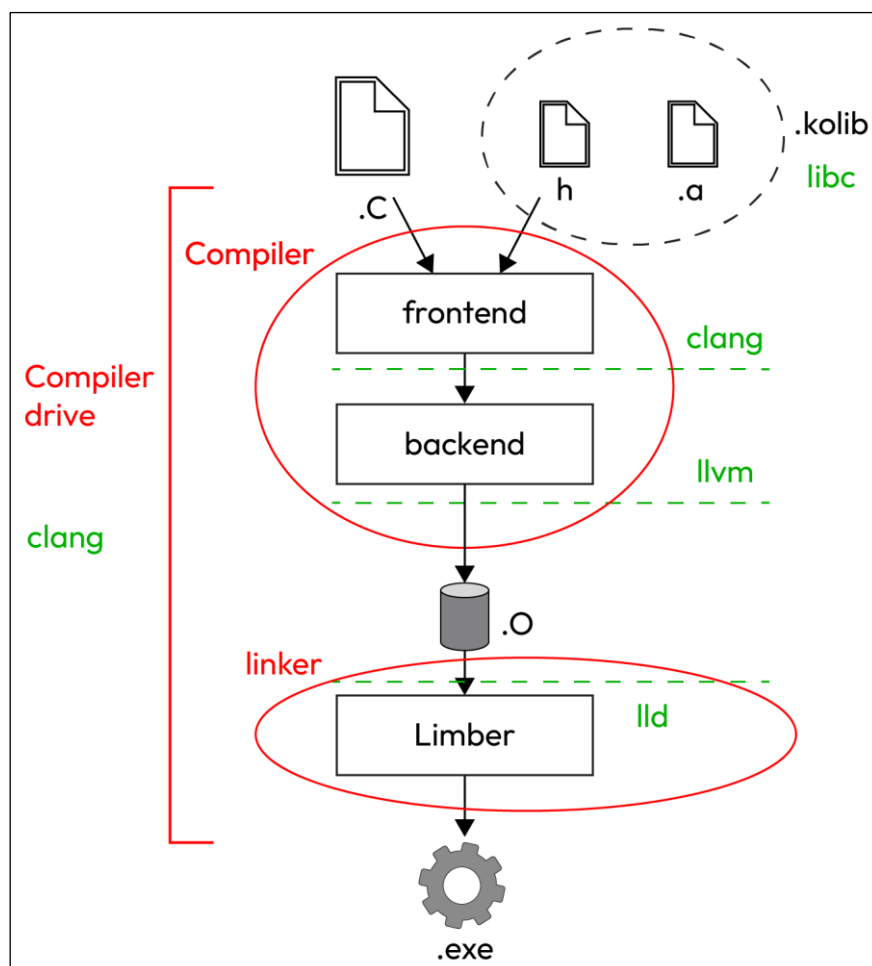
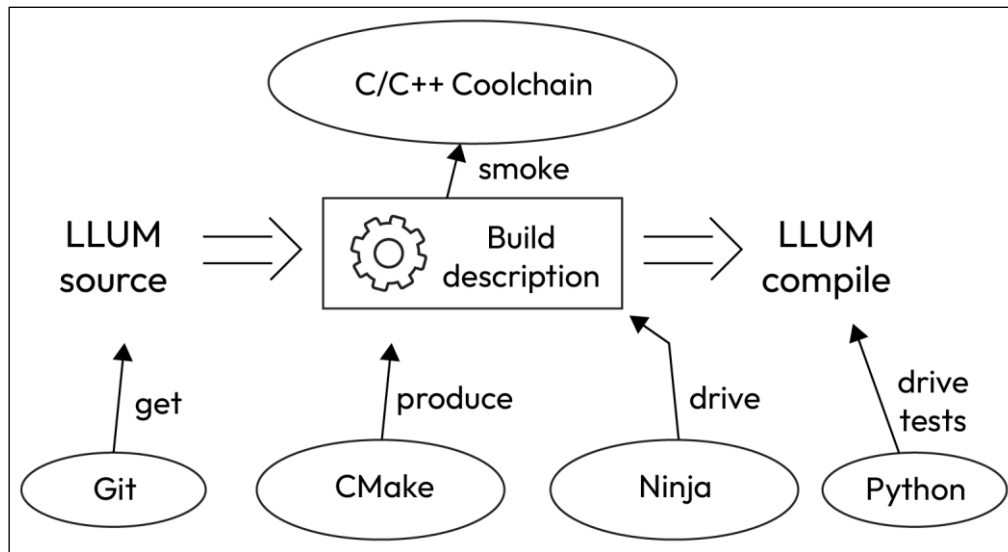
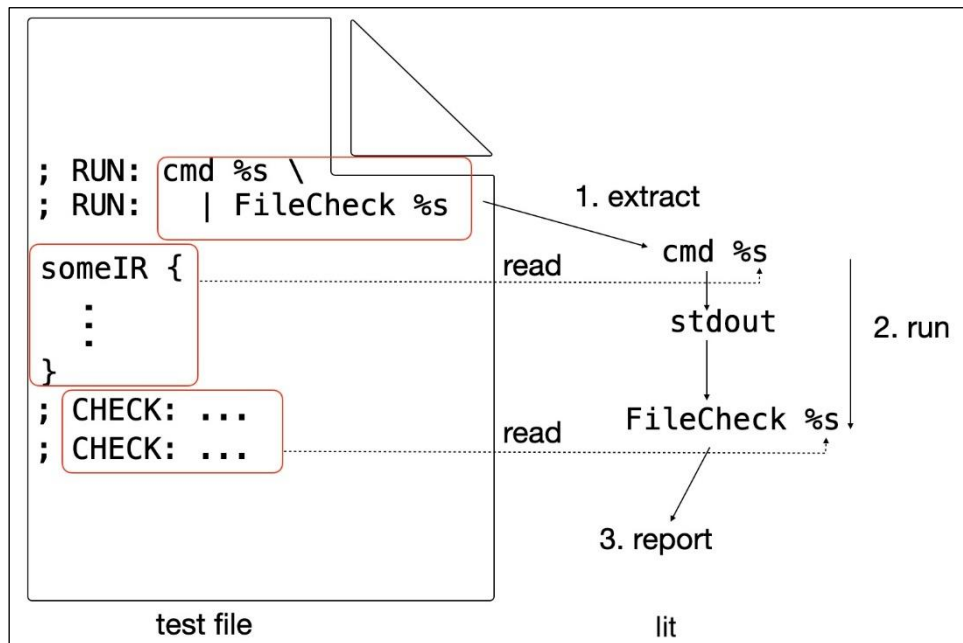
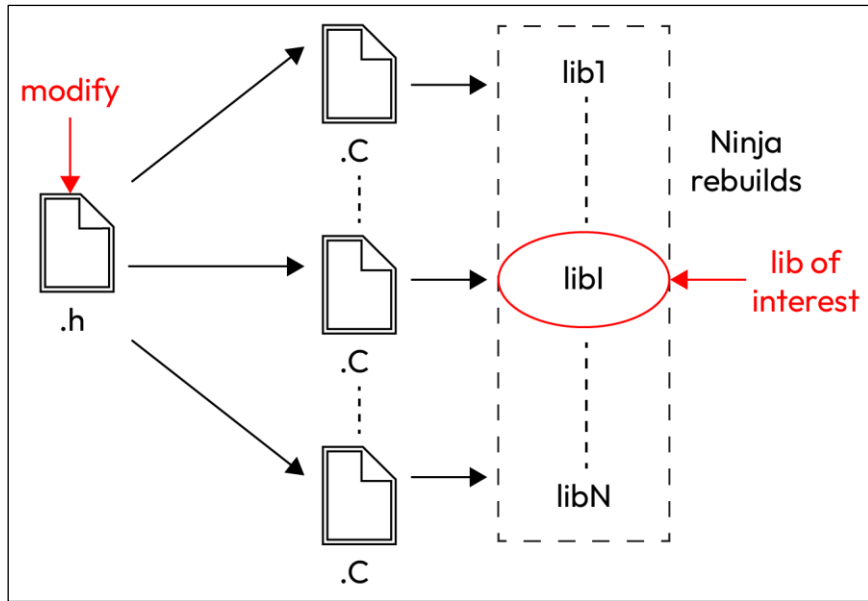
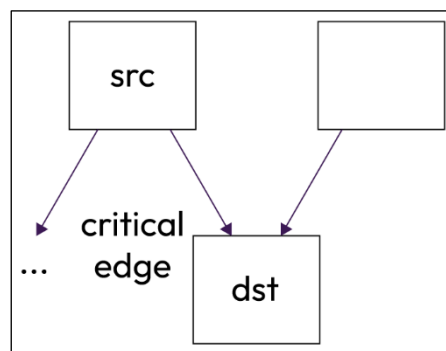
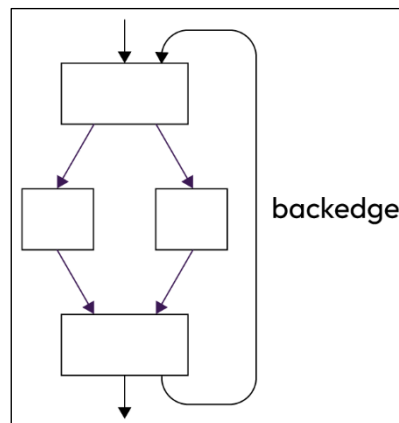
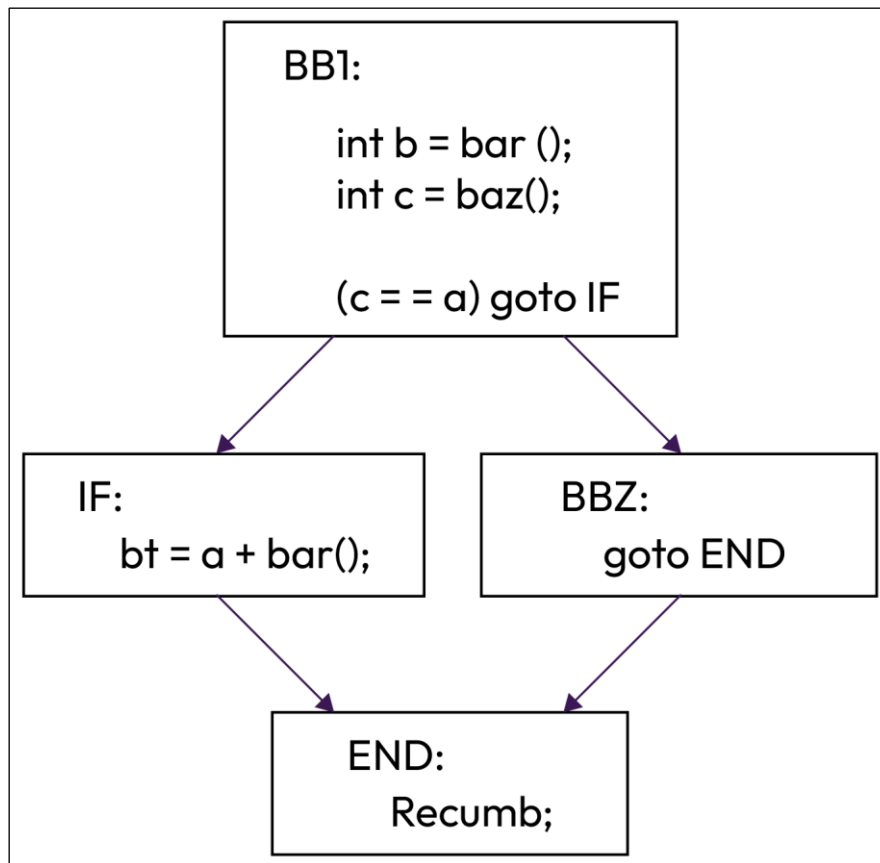


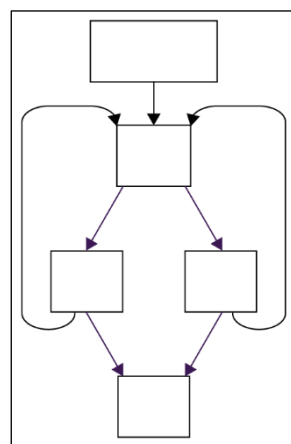
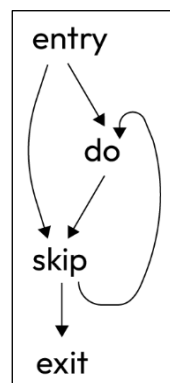
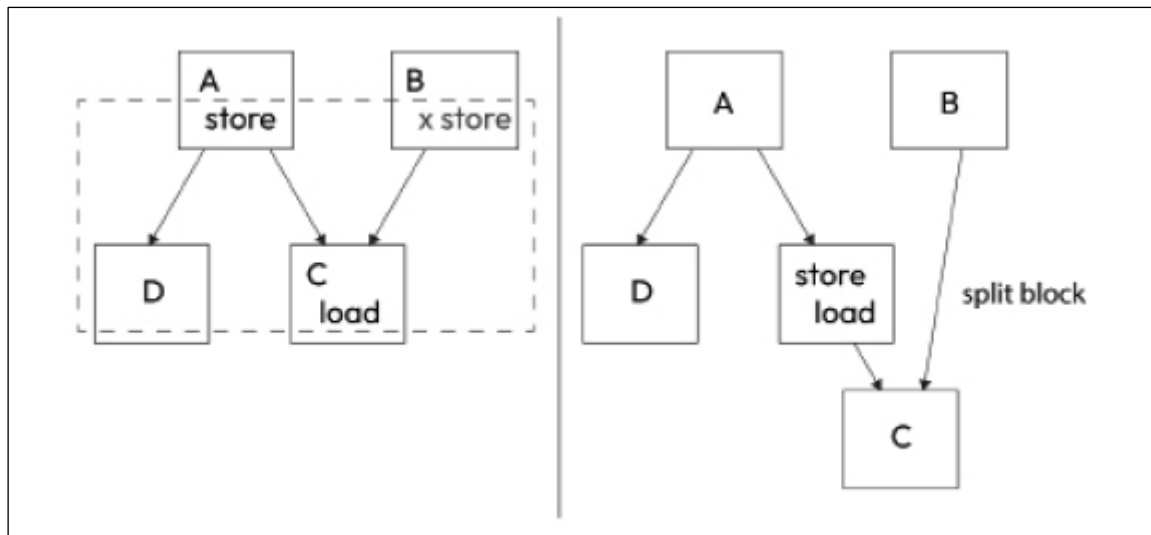
Chapter 1: Building LLVM and Understanding the Directory Structure

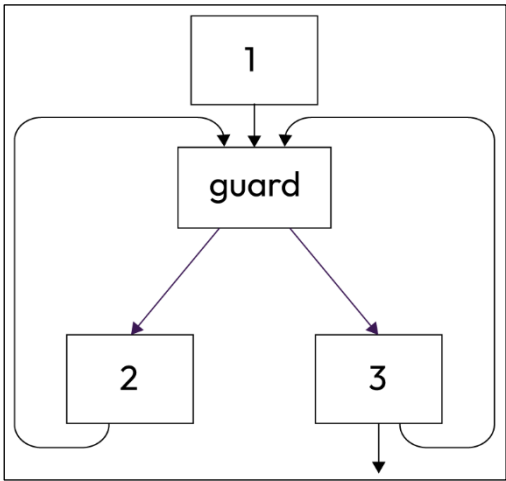
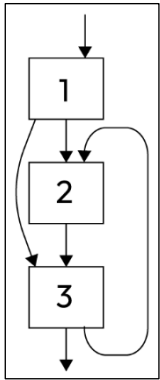
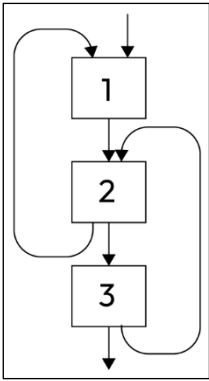




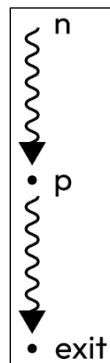
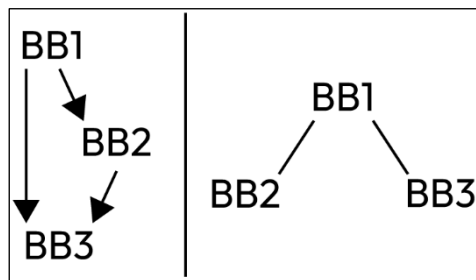
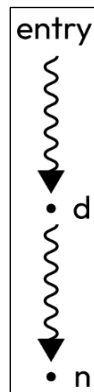
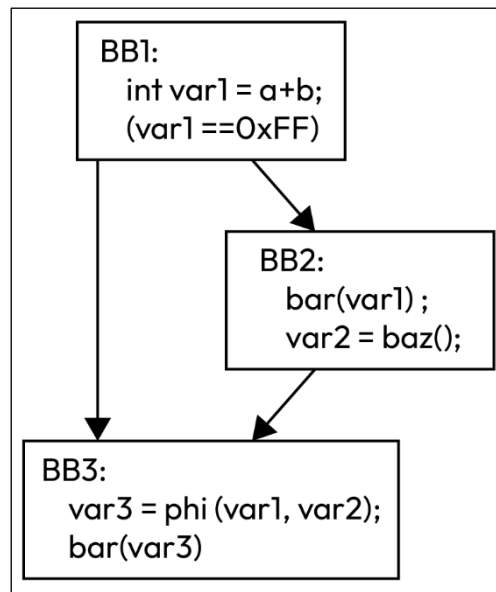
Chapter 3: Compiler Basics and How They Map to LLVM APIs

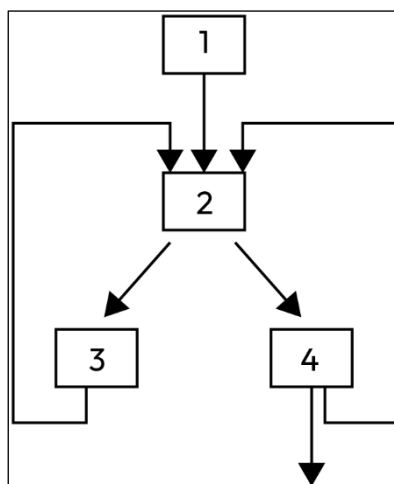
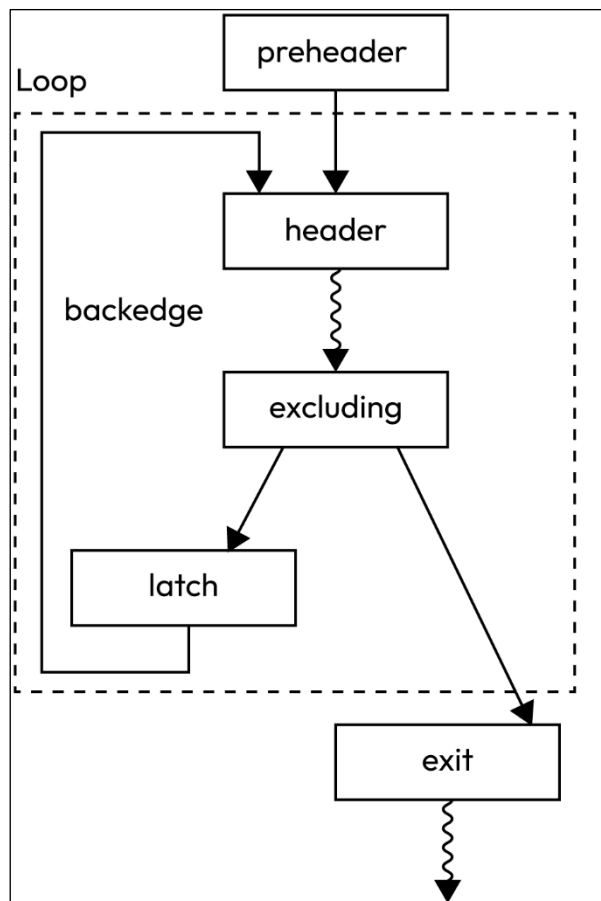
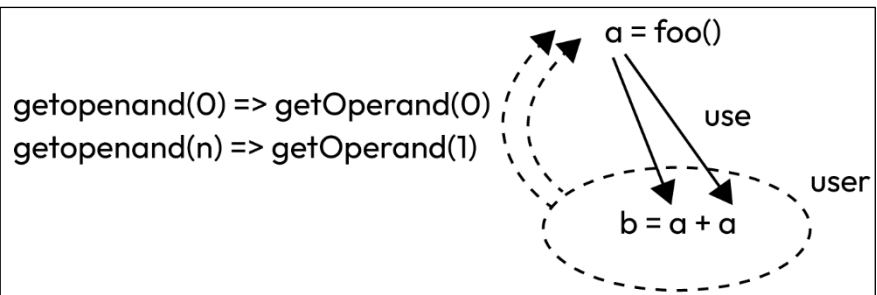




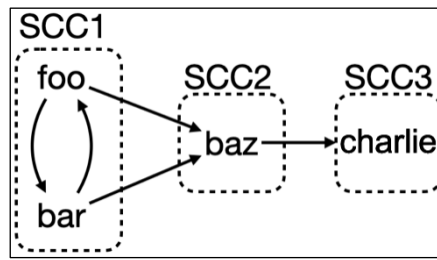


Chapter 4: Writing Your First Optimization

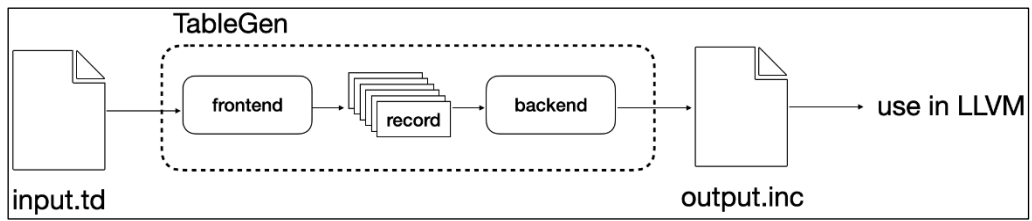




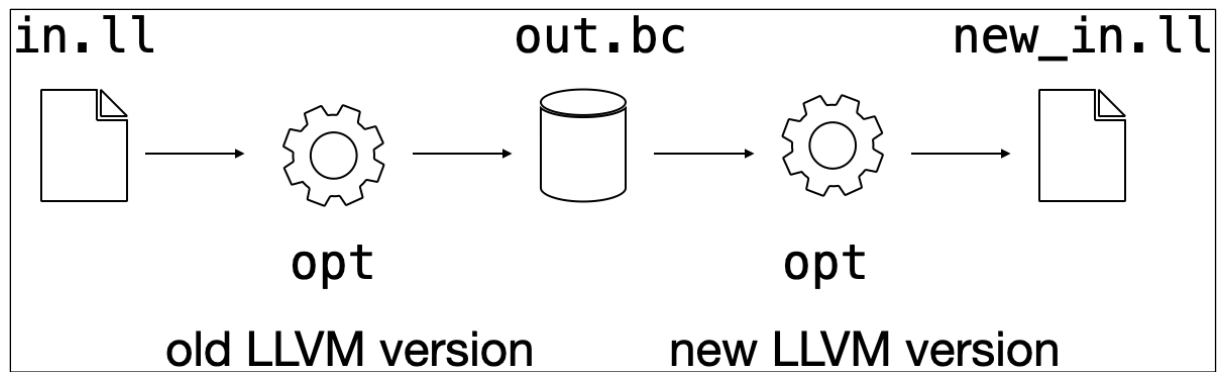
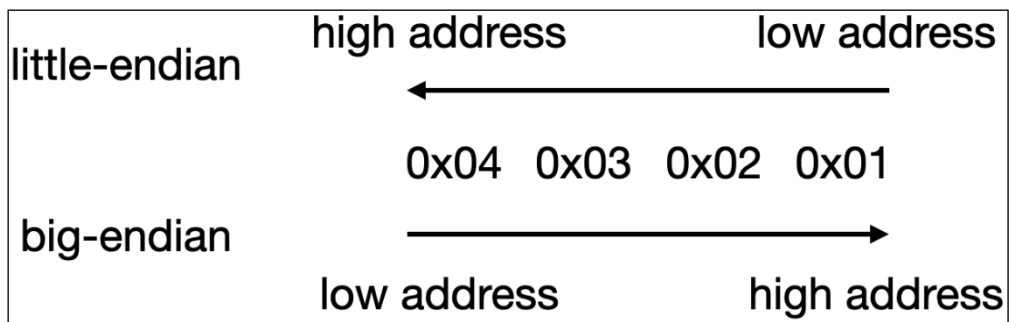
Chapter 5: Dealing with Pass Managers

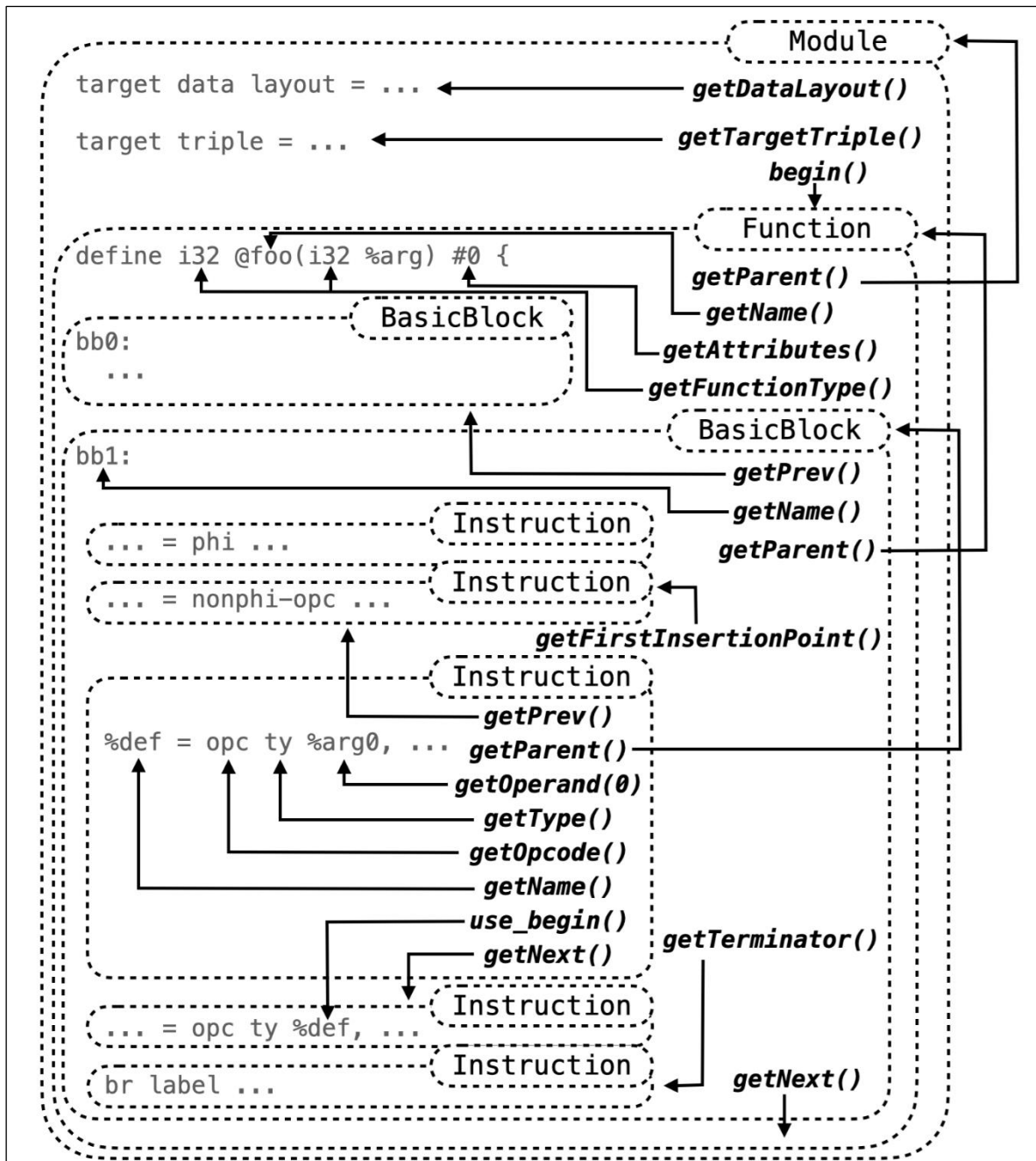


Chapter 6: TableGen – LLVM Swiss Army Knife for Modeling

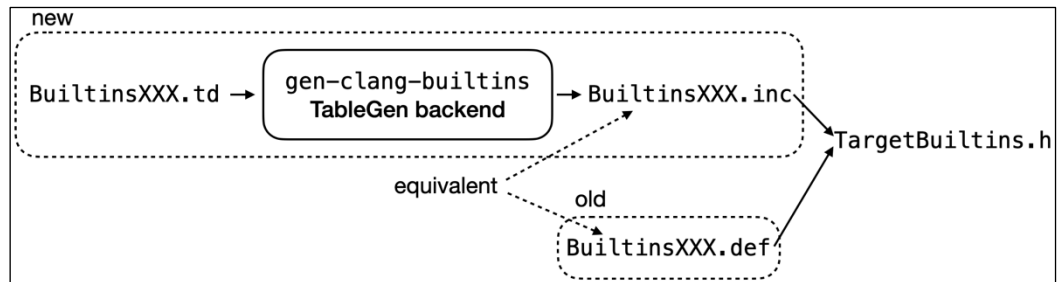
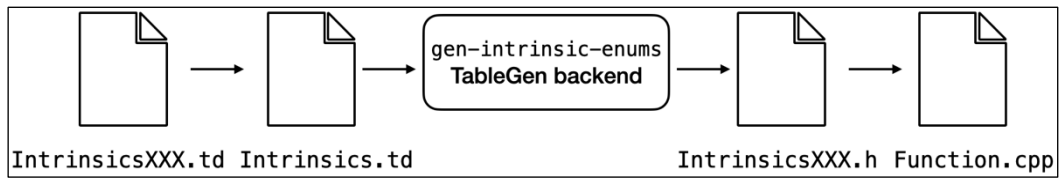
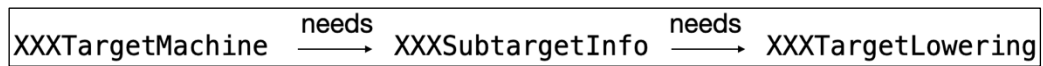


Chapter 7: Understanding LLVM IR

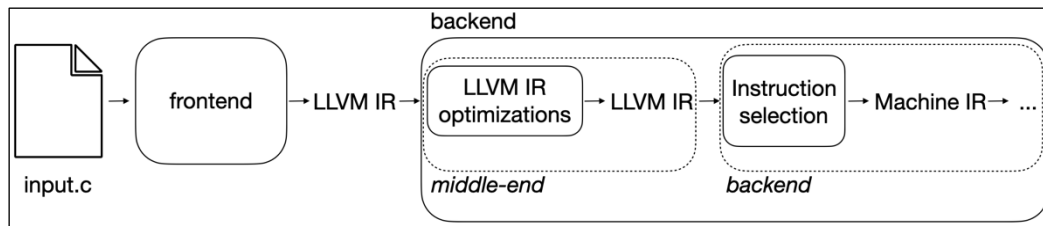




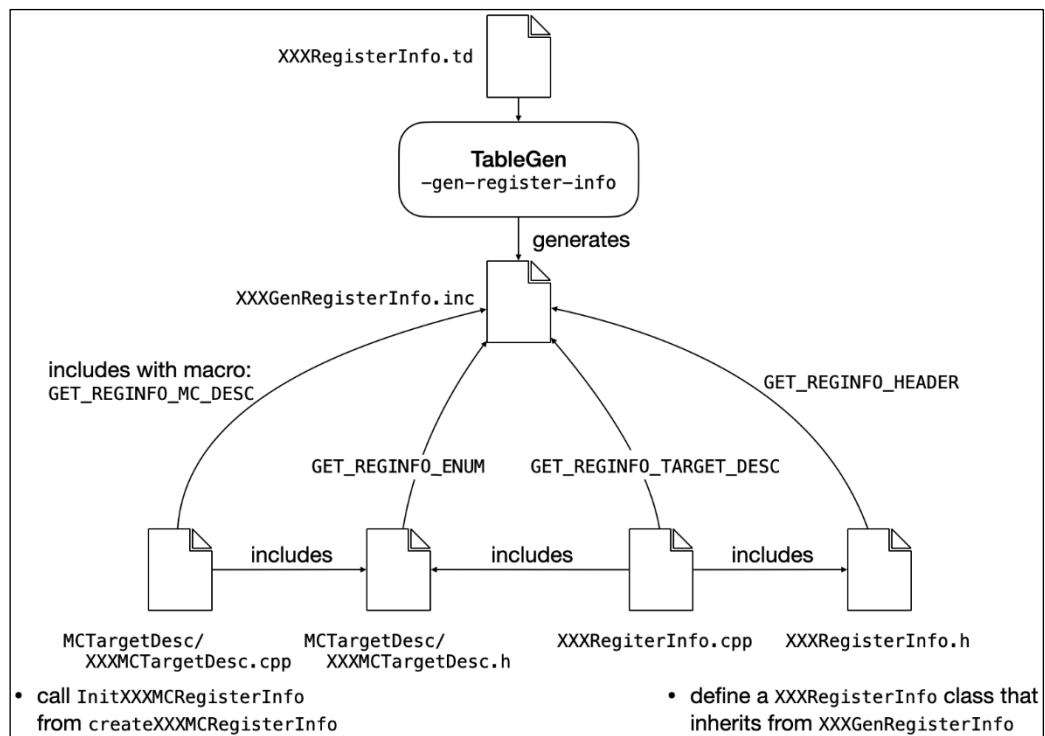
Chapter 9: Introducing Target-Specific Constructs

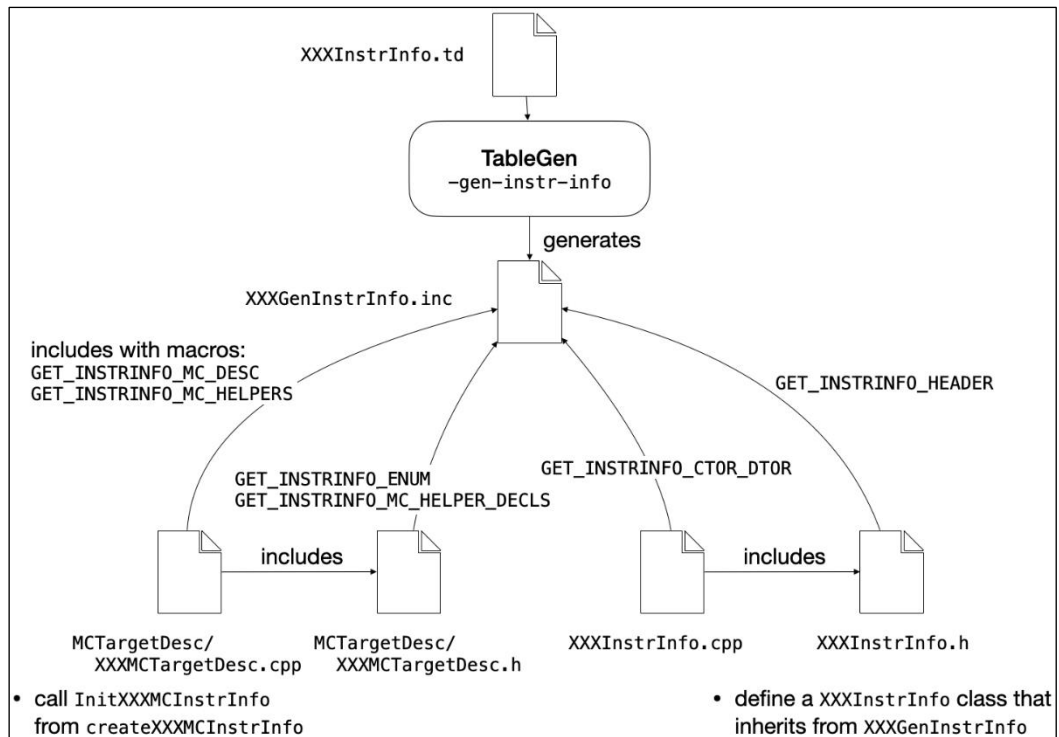


Chapter 11: Getting Started with the Backend

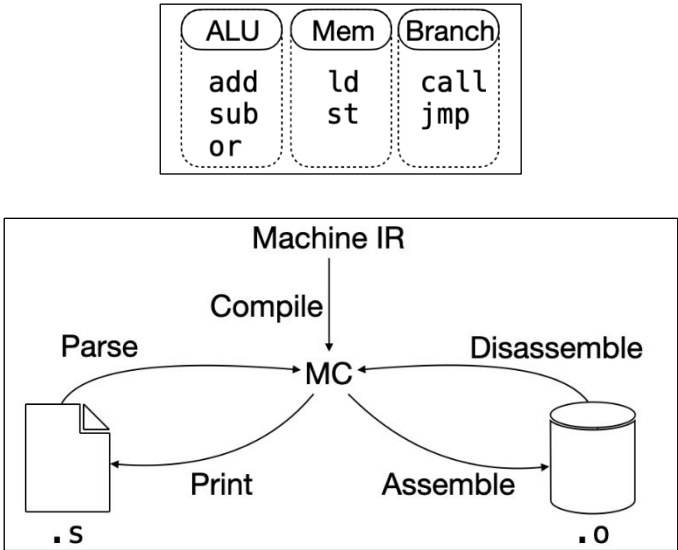


128-bit quad	q0			q1		q2	
64-bit double	d0	d1		d2			
32-bit single	s0	s1	s2				
unit	u0	u1	u2	u3		u4	





Chapter 12: Getting Started with the Machine Code Layer



SME FP16 widening outer product

31

29

21|20

16|15

13|12

10|9

5|4

3|2|1

0

1

0|0

0

0

0

0

0

1

1

0

1

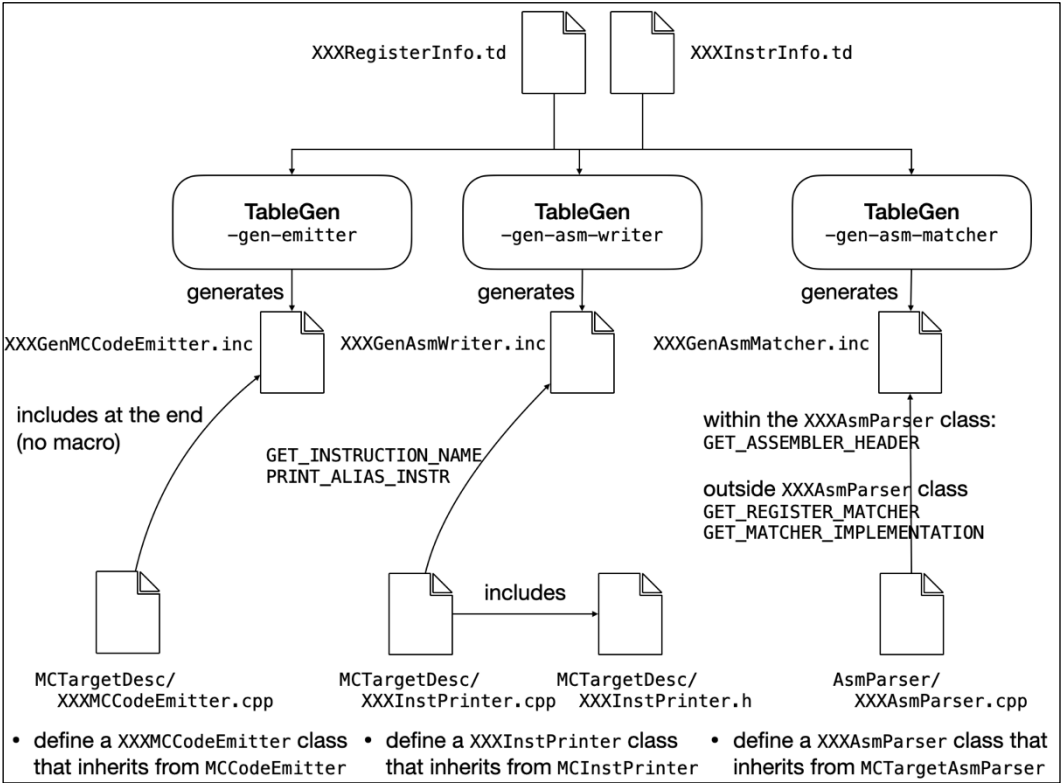
Zm

Pm

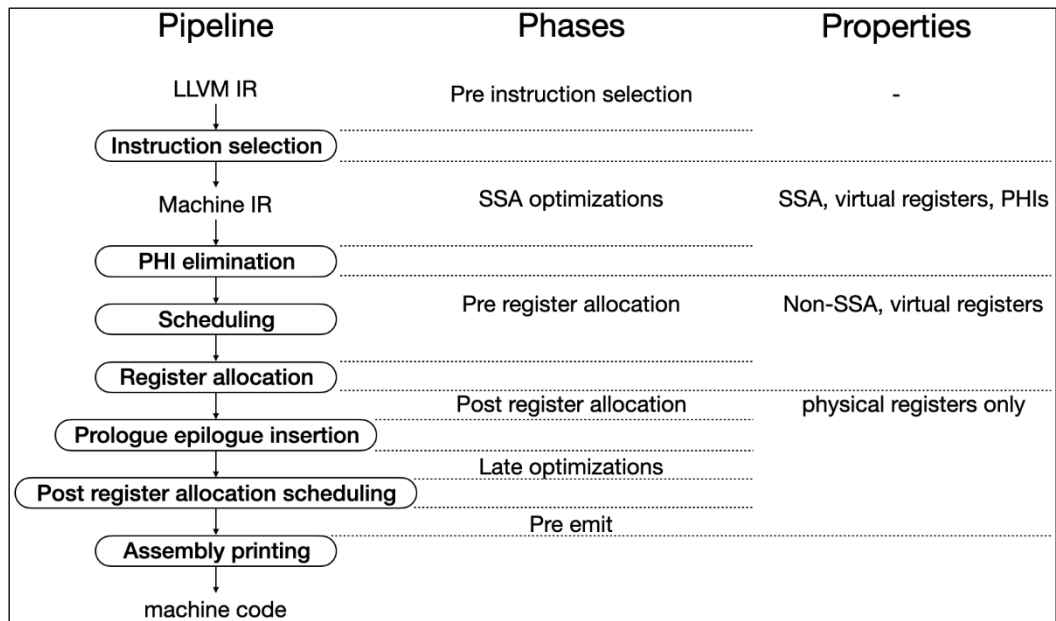
Pn

Zn

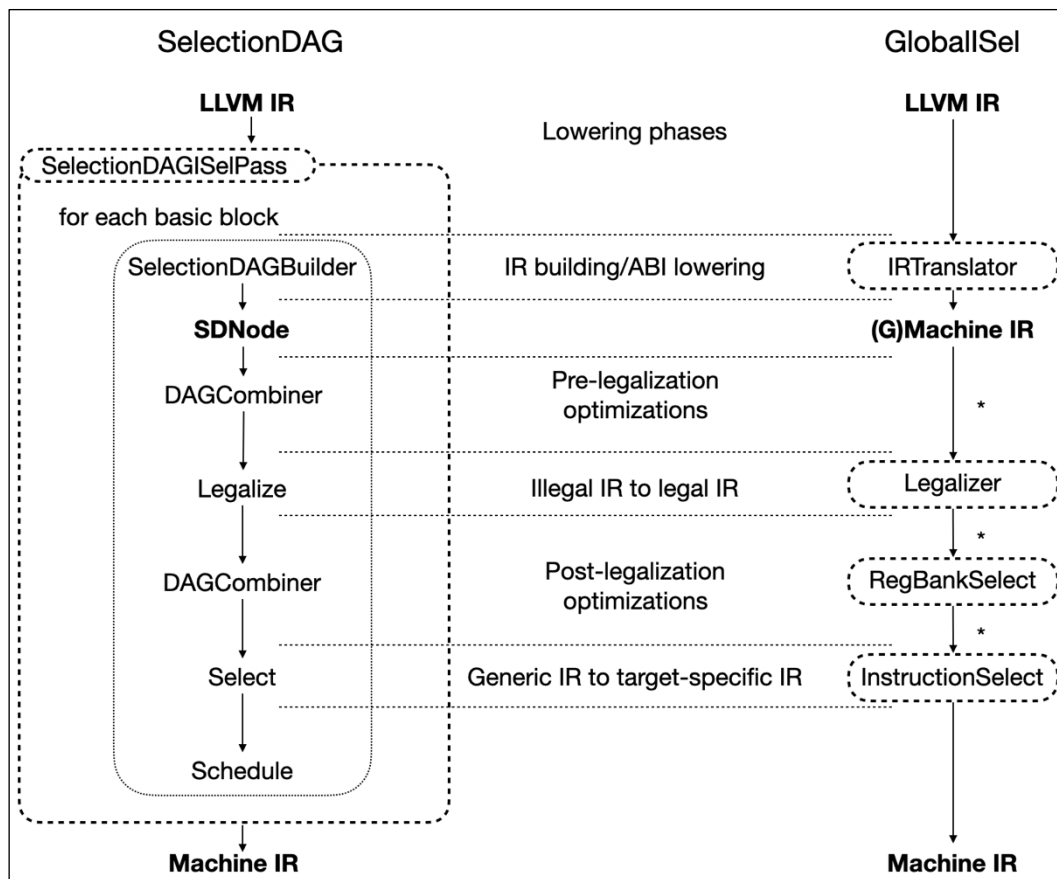
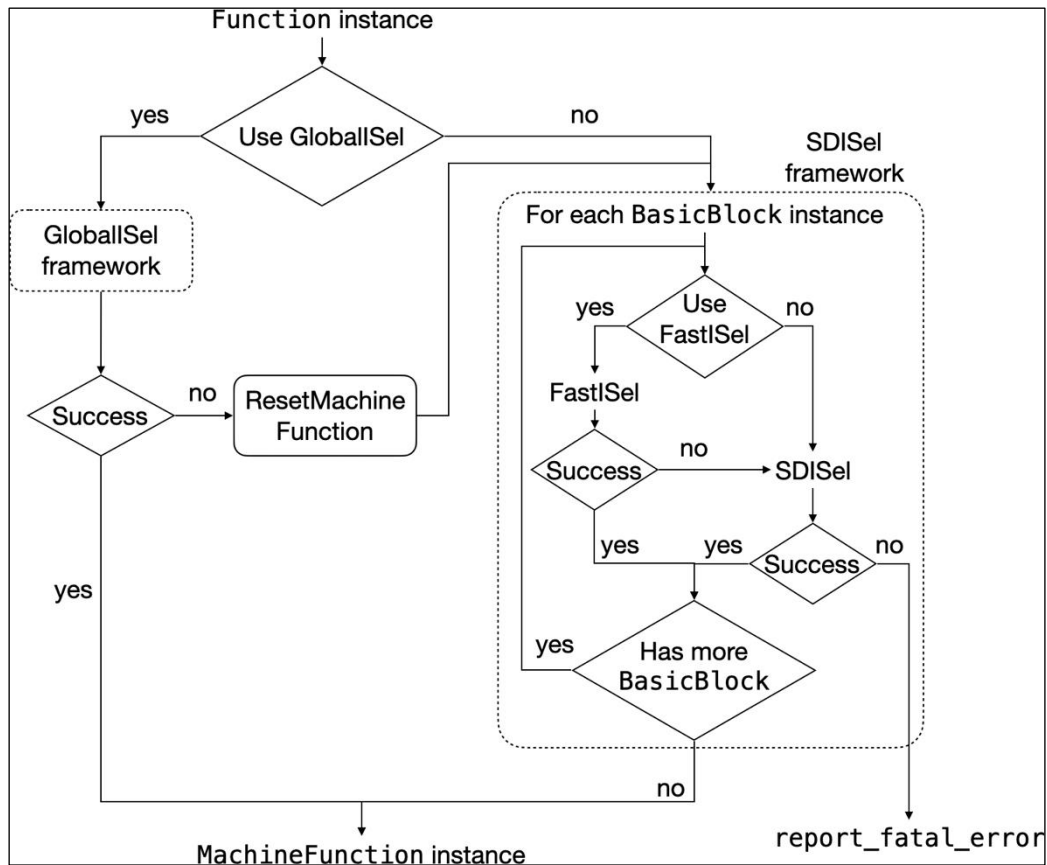
S|0|0|ZAda



Chapter 13: The Machine Pass Pipeline

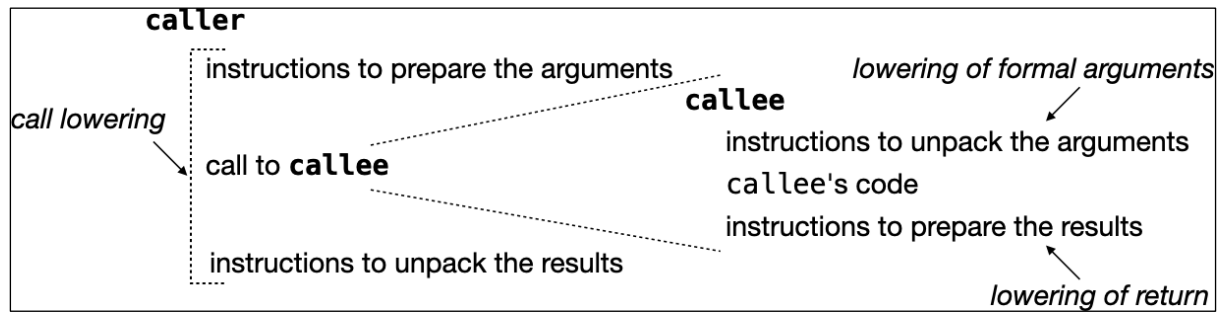


Chapter 14: Getting Started with Instruction Selection



Pass Pipeline	Generic MachineInstr	Generic virtual register		MachineFunction Properties		
LLVM IR ↓ IRTranslator ↓ (G)Machine IR ↓ Legalizer ↓ RegBankSelect ↓ InstructionSelect ↓ Machine IR		Type	RegisterBank	legalized	regBankSelected	selected
	✓	✓	Optional	false	false	false
	✓ Only legal opcodes	✓ Only legal types	Optional	true	false	false
	✓ Only legal opcodes	✓ Only legal types	Must	true	true	false
	✗	✗	✗	true	true	true

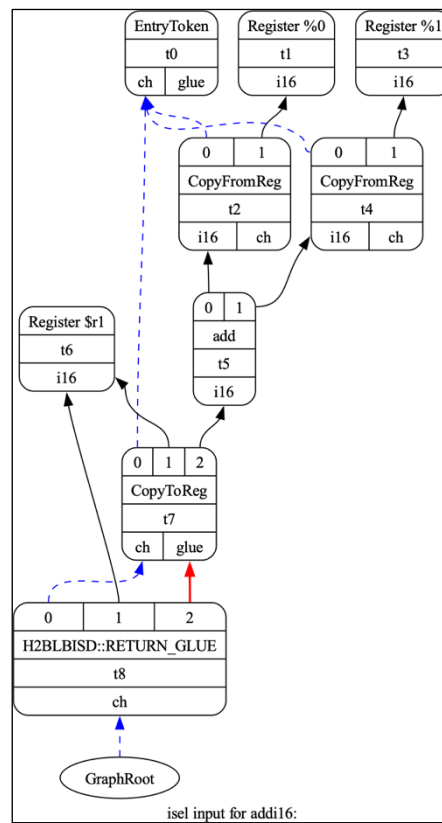
Chapter 15: Instruction Selection: The IR Building Phase



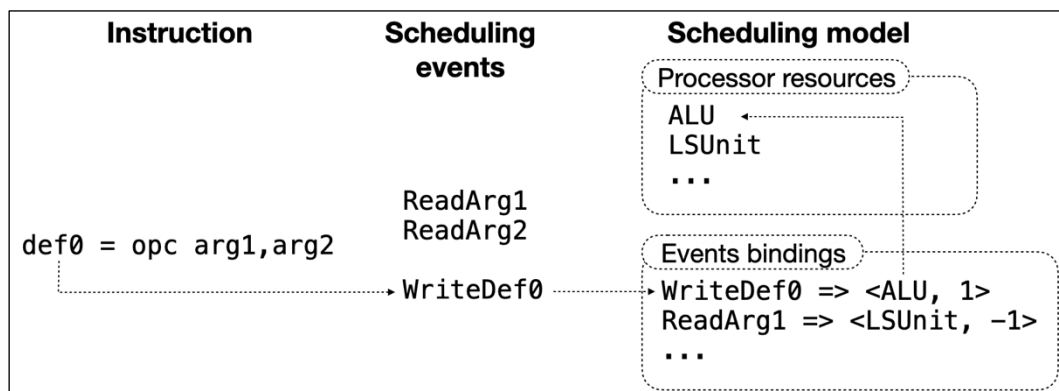
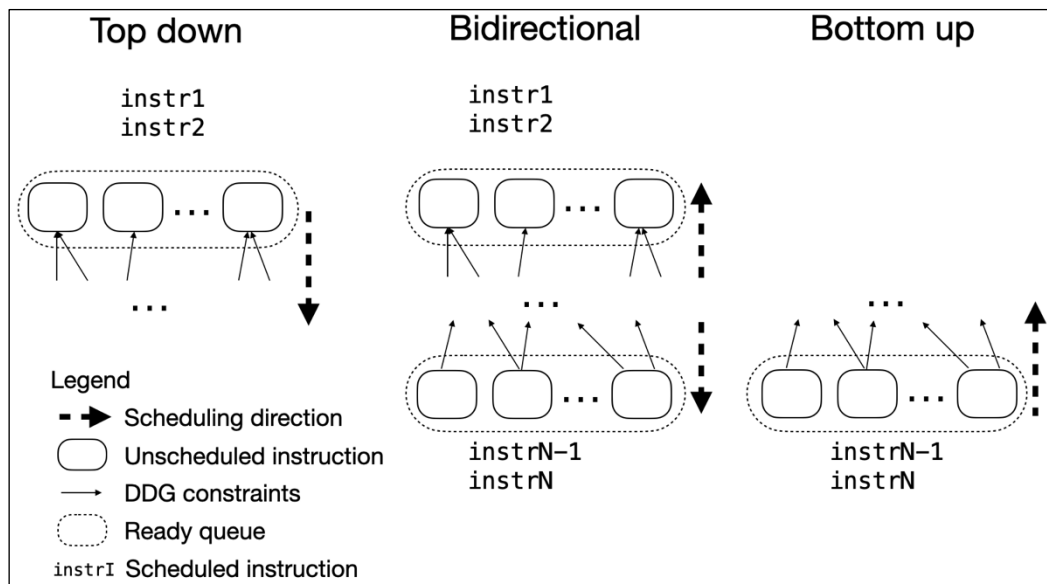
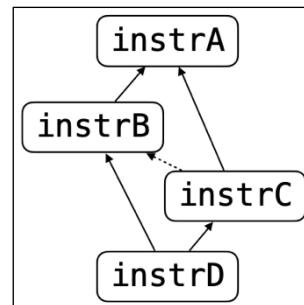
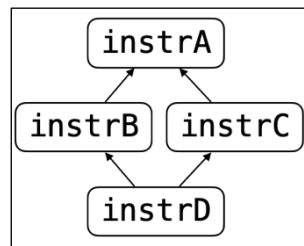
Chapter 17: Instruction Selection: The Selection Phase and Beyond

```
%0:_(<2 x s32>) = G_AND %1, %2
```

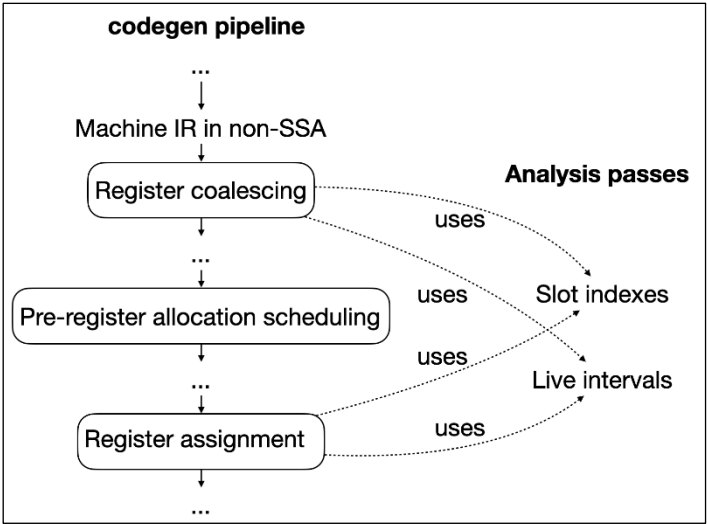
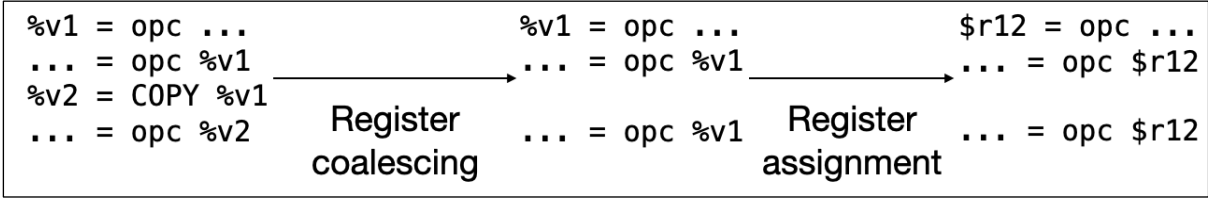
Vector InstructionMapping			Scalar InstructionMapping		
Operand index	ValueMapping		Operand index	ValueMapping	
	#Partial values	PartialMapping		#Partial values	PartialMapping
0	1	{0, 64, VR}	0	2	{0, 32, GPR} {32, 32, GPR}
1	1	{0, 64, VR}	1	2	{0, 32, GPR} {32, 32, GPR}
2	1	{0, 64, VR}	2	2	{0, 32, GPR} {32, 32, GPR}



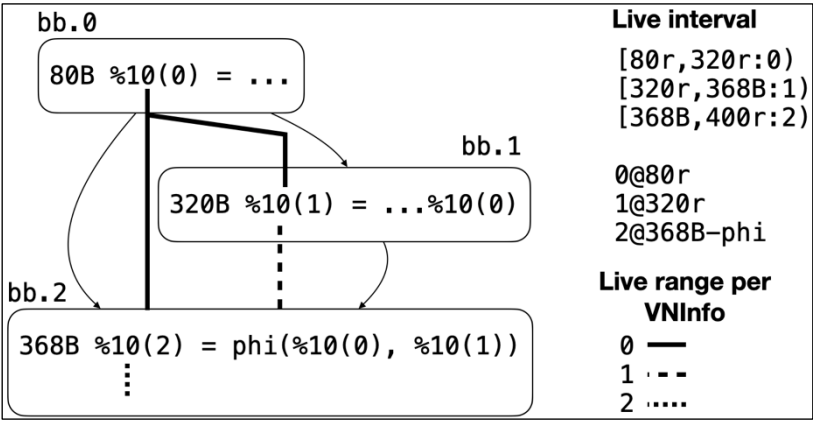
Chapter 18: Instruction Scheduling



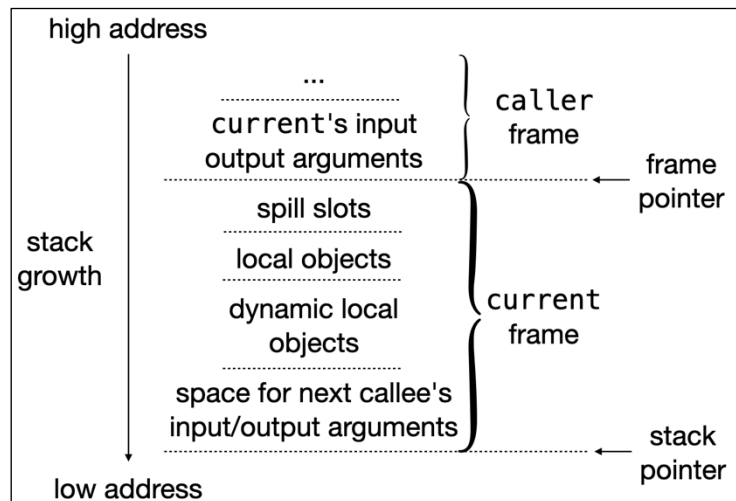
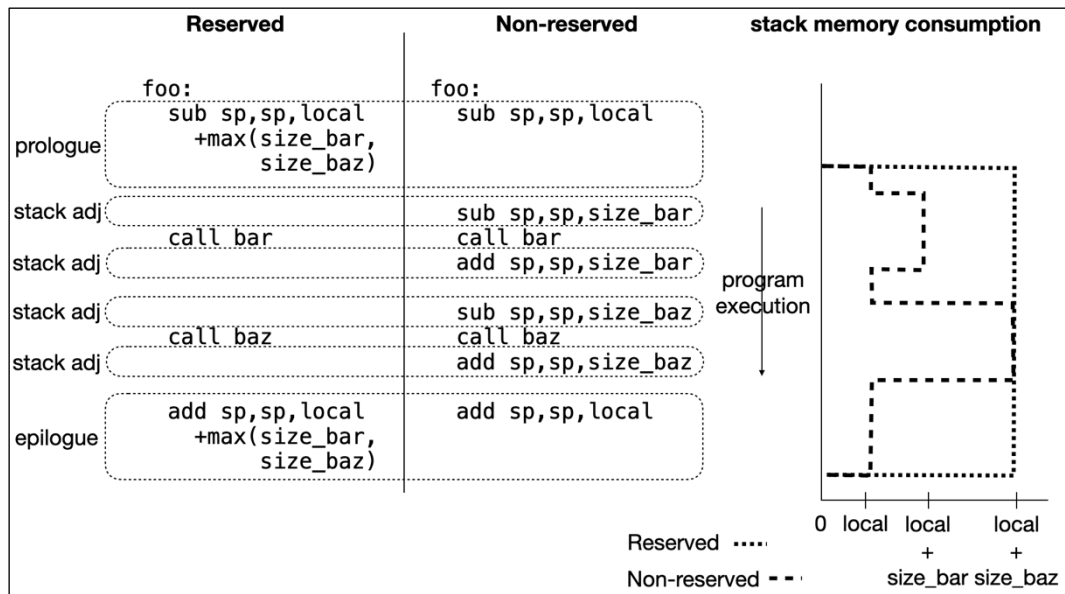
Chapter 19: Register Allocation



SlotIndex		MachineInstr
Index	Slot	
16	block(B)	
	early-clobber(e)	
	register(r)	... = instrI
	dead(d)	
32	B	
	e	
	r	def = instrJ arg
	d	



Chapter 20: Lowering of the Stack Layout



Chapter 21: Getting Started with the Assembler

