SWPP Backend Assembly Language (Tentative)

- All registers are 64 bits, has no type. Register names should be r1.. r16. Function arguments have its own read-only registers (arg1.. arg16).
- No SSA, No PHI node.
- Accessing stack is done via load/store to alloca.

1. Memory operations

Kind	Name	Cost
Stack Allocation	alloca	1
Heap Allocation	malloc	1
Deallocation	free	1
Load	load*	Stack area: 2 Heap area: 4
Store	store*	Stack area: 2 Heap area: 4
Reset tape access pin	reset [stack|heap]	2

- Load and store has an additional cost for moving the pin to the place to record.
cost = 0.004 * (cur addr - prev addr)
- Stack area starts from 1024, grows downward (-)

- Heap area stars from 2048, grows upward (+)
- Malloc finds an empty space with smallest address in heap area & allocates it.
- Memory consumption cost will be measured (TODO)

2. Non-memory operations

Kind	Name	Cost
Terminator	Ret Br Switch	1
Integer Multiplication/Division	UDiv SDiv	0.8

	URem SRem Mul Shl LShr AShr	
Integer Add/Sub/Logical Ops	Add Sub And Or Xor	1.2
Floating Pt Mul/Div	FMul FDiv	0.8
Floating Pt Add/Sub	FAdd FSub	1.2
Comparison	ICmp FCmp	1
Else	Select Call	1.2 2

There is no callee save / caller save registers