ZanaX Reference Data

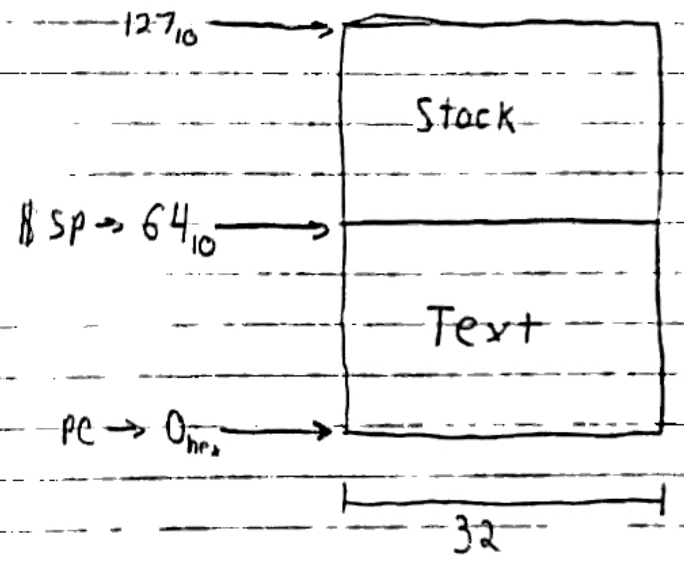
Name:		format	operation	opcode/funct (hex)
Add	add	R	R[rd] = R[rs] + R[rt]	$0/3_{hex}$
Add immediate	addi	I	R[rt] = R[rs] + SignExtImm	3_{hex}
And	and	R	R[rd] = R[rs] & R[rt]	$0/0_{hex}$
Or	or	R	R[rd] = R[rs] | R[rt]	$0/1_{hex}$
NOr	nor	R	R[rd] = ~(R[rs] | R[rt])	$0/2_{hex}$
Subtroct	sub	R	R[rd] = R[rs] - R[rt]	$0/4_{hex}$
Mult	mult	R	{Hi,Lo} = R[rs] * R[rt]	$0/5_{hex}$
Set Less Than	slt	R	R[rd] = (R[rs] < R[rt])? 1:0	$0/6_{hex}$
Jump Register	Jr	R	PC = R[rs]	$0/7_{hex}$
Load Word	lw	I	R[rt] = M[R[rs] + SignExtImm]	1_{hex}
Store Word	sw	I	M[R[rs] + SignExtImm] = R[rt]	2_{hex}
Subtroct Immediate	subi	I	R[rt] = R[rs] - SignExtImm	4_{hex}
Branch on Equal	beq	I	if (R[rs] == R[rt]) PC = PC + 4 + BranchAddr	5_{hex}
Branch on Not Equal	bne	I	if (R[rs] != R[rt]) PC = PC + 4 + BranchAddr	9_{hex}
Jump	J	J	PC = JumpAddr	7_{hex}
Jump and Link	Jal	J	R[15] = PC + 8; PC = JumpAddr	8_{hex}

Instruction Formats

	31 26	25 21	20 16	15 11	10 6	5 0
R	op	rs	rt	rd	shamt	funct
I	op	rs	rt	immediate		
J	op	address				

Register Name	Number	Use
$zero	0	Constant zero reg.
$gpr1 - $gpr4	1-4	general purpose registers
$a0 - $a2	5-7	Argument Registers
$t0 - $t3	8-11	Temporary Registers
$v0 - $v1	12-13	Value Registers for function results
$sp	14	Stack Pointer Register
$ra	15	Return Address Register
$gpr16 - $gpr31	16-31	More general purpose reg's.

MEMORY ALLOCATION


127_{10}
Stack
$sp → 64_{10}
Text
PC → 0_{hex}
32