

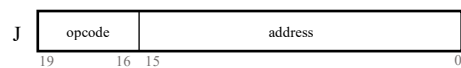
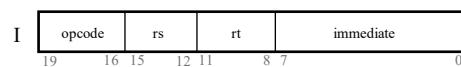
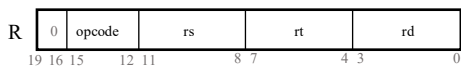
# CAGD Reference Data



## CORE INSTRUCTION SET

NAME, MNEMONIC	FOR-MAT	OPERATION	OPCODE
Add	add R	$R[rd] = R[rs] + R[rt]$	0000
Addi	addi I	$R[rd] = R[rs] + 0$	0001
Subtract	sub R	$R[rd] = R[rs] - R[rt]$	0010
Multiply	mul R	$R[rd] = R[rs] * R[rt]$	0011
Branch On Equal	beq I	If( $R[rs] == R[rt]$ ) PC=BranchAddr	0100
Branch On Not Equal	bnq I	If( $R[rs] != R[rt]$ ) PC=BranchAddr	0101
Branch Less Than	blt I	If( $R[rs] < R[rt]$ ) PC=Label	0110
Branch Greater Than	bgt I	If( $R[rs] > R[rt]$ ) PC=Label	0111
Branch Greater Than Or Equal	bgte I	If( $R[rs] \geq R[rt]$ ) PC=Label	1000
Branch Less Than Or Equal To Zero	bgtz I	If( $R[rs] < R[rt]$ ) PC=Label	1001
Jump	j J	PC = JumpAddr	1010
Input	in I	J	1011
Output	out I		1100
Load Word	lw I		1101
Save word	sw I		1110

## BASIC INSTRUCTION FORMATS



## REGISTER NAME, NUMBER, USE

NAME	NUMBER	USE
\$r0-\$r7	0-7	Temporaries
\$at	8	Assembler Temporary
\$ip	9	Pointer for Input
\$op	10	Pointer for Output
\$gp	11	Global Pointer
\$0	12	Zero

## OPCODES, BASE CONVERSION, ASCII SYMBOLS

CAGD Opcode (15:12)	Binary	Decimal	Hexadecimal	ASCII Character
add	0000	0	0	NUL
addi	0001	1	1	SOH
sub	0010	2	2	STX
mul	0011	3	3	ETX
beq	0100	4	4	EOT
bnq	0101	5	5	ENQ
blt	0110	6	6	ACK
bgt	0111	7	7	BEL
bgte	1000	8	8	BS
bgtz	1001	9	9	HT
j	1010	10	a	LF
in	1011	11	b	VT
out	1100	12	c	FF

## MEMORY ALLOCATION

