RISC-V (RARS) Reference Card

Instructions

```
# S - signed
# U - unsigned
# P - pseudo-instruction

# Arithmetic
add     t1, t2, t3
addi    t1, t2, -100
sub     t1, t2, t3
neg     t1, t2        # P

# Logical
and     t1, t2, t3
andi    t1, t2, -100
or      t1, t2, t3
ori     t1, t2, -100
xor     t1, t2, t3
xori    t1, t2, -100
not     t1, t2        # P

# Shifts
sll     t1, t2, t3    # left logical
slli    t1, t2, 33    # left logical
sra     t1, t2, t3    # right arithmetic (S)
srai    t1, t2, 33    # right arithmetic (S)
srl     t1, t2, t3    # right logical (U)
srli    t1, t2, 33    # right logical (U)

# Multiplication
mul     t1, t2, t3    # t1 <- t2*t3[31:0]
mulh    t1, t2, t3    # t1 <- t2*t3[63:32] (S)
mulhu   t1, t2, t3    # t1 <- t2*t3[63:32] (U)
mulhsu  t1, t2, t3    # t1 <- t2*t3[63:32] (t2 S, t3 U)

# Division, remainder
div     t1, t2, t3    # S
divu    t1, t2, t3    # U
rem     t1, t2, t3    # S
remu    t1, t2, t3    # U

# Load value from memory at (t2-100) to t1
lb      t1, -100(t2)  # sign-extended 8-bit
lbu     t1, -100(t2)  # zero-extended 8-bit
lh      t1, -100(t2)  # sign-extended 16-bit
lhu     t1, -100(t2)  # zero-extended 16-bit
lw      t1, -100(t2)  # 32-bit

# Store value t1 to memory at (t2-100)
sb      t1, -100(t2)  # 8-bit
sh      t1, -100(t2)  # 16-bit
sw      t1, -100(t2)  # 32-bit

# System
ecall
ebreak

# Other
lui     t1, imm       # t1 <- imm << 12
auipc   t1, imm       # t1 <- pc + (imm << 12)
mv      t1, t2        # t1 <- t2 (P)
li      t1, 1000      # t1 <- 1000 (P)
la      t1, label     # t1 <- label (P)
nop                   # no operation (P)
```

Registers

Register	ABI name	Saver
x0	zero	--
x1	ra	Caller
x2	sp	Callee
x3	gp	--
x4	tp	Callee
x5-x7	t0-t2	Caller
x8	s0/fp	Callee
x9	s1	Callee
x10-x17	a0-a7	Caller
x18-x27	s2-s11	Callee
x28-x31	t3-t6	Caller

```
# Branches
beq     t1, t2, target   # if t1 == t2
bne     t1, t2, target   # if t1 != t2
blt     t1, t2, target   # if t1 < t2 (S)
bltu    t1, t2, target   # if t1 < t2 (U)
bgt     t1, t2, target   # if t1 > t2 (S) (P)
bgtu    t1, t2, target   # if t1 > t2 (U) (P)
ble     t1, t2, target   # if t1 <= t2 (S) (P)
bleu    t1, t2, target   # if t1 <= t2 (U) (P)
bge     t1, t2, target   # if t1 >= t2 (S)
bgeu    t1, t2, target   # if t1 >= t2 (U)
beqz    t1, target       # if t1 == 0 (P)
bnez    t1, target       # if t1 != 0 (P)
bltz    t1, target       # if t1 < 0 (P)
bgtz    t1, target       # if t1 > 0 (P)
blez    t1, target       # if t1 <= 0 (P)
bgez    t1, target       # if t1 >= 0 (P)

# Comparisons
slt     t1, t2, t3       # t1 <- t2 < t3 (S)
sltu    t1, t2, t3       # t1 <- t2 < t3 (U)
slti    t1, t2, -100     # t1 <- t2 < -100 (S)
sltiu   t1, t2, -100     # t1 <- t2 < -100 (U)
sgt     t1, t2, t3       # t1 <- t2 > t3 (S) (P)
sgtu    t1, t2, t3       # t1 <- t2 > t3 (U) (P)
seqz    t1, t2           # t1 <- t2 == 0 (P)
snez    t1, t2           # t1 <- t2 != 0 (P)
sltz    t1, t2           # t1 <- t2 < 0 (P)
sgtz    t1, t2           # t1 <- t2 > 0 (P)

# Jump and link
jal     t1, target       # t1 <- pc+4; pc = target
jal     target           # ra <- pc+4; pc = target (P)
j       target           # pc = target (P)
b       target           # pc = target (P)
jalr    t1, t2, -100     # t1 <- pc+4; pc = t2-100
jalr    t2, -100         # ra <- pc+4; pc = t2-100 (P)
jalr    t2               # ra <- pc+4; pc = t2 (P)
jr      t2, -100         # pc = t2-100 (P)
jr      t2               # pc = t2 (P)
ret                      # pc = ra (P)
```

Directives

```
# code section
.text
# data section
.data

.byte   x
.half   x
.word   x
.dword  x
.float  x
.double x
```

```
# align to 2^n
.align  n
# reserve n bytes
.space  n
# chars
.ascii  "abc"
# zero-term. chars
.asciz  "abc"
# alias for .asciz
.string "abc"
```

```
.globl  f

.eqv    N, 10

.include "abc.asm"

.macro
.end_macro
```

Sys. calls

1	PrintInt	11	PrintChar	40	RandSeed
2	PrintFloat	12	ReadChar	41	RandInt
3	PrintDouble	17	GetCWD	42	RandIntRange
4	PrintString	30	Time	43	RandFloat
5	ReadInt	31	MidiOut	44	RandDouble
6	ReadFloat	32	Sleep	57	Close
7	ReadDouble	33	MidiOutSync	62	LSeek
8	ReadString	34	PrintIntHex	63	Read
9	Sbrk	35	PrintIntBinary	64	Write
10	Exit	36	PrintIntUnsigned	93	Exit2