# BETA ISA

A：0、1、2、3、0xc

B：0x2000、0xEDEDEDED、0xFEDEDEDE、0x2004、110000 00000 11111 0010 0000 0000 0000（0xc01f2000）

C：0x87654321、1、0x87654320、0x14、011101 11111 00001 0000 0000 0000 0010(0x77e10002)

D：17、0、32、no instructions need to be changed

E：4、110000 00000 00000 0000 0000 0000 0000（0xC0000000）

# 快速排序

A：0x14

B：0、-1

C：0xa

D：first (smaller elements)

E：0、4

F：5

p\_array=R2 // base address of array (arg 0)

p\_left=R3

p\_right=R4

p\_pivotIndex=R5 // Corresponds to PivotIndex in C program

p\_pivotValue=R6

p\_storeIndex=R7

partition:

//.breakpoint

PUSH(LP)

PUSH(BP)

MOVE(SP, BP)

PUSH(R1)

PUSH(R2)

PUSH(R3)

PUSH(R4)

PUSH(R5)

PUSH(R6)

PUSH(R7)

PUSH(R8)

PUSH(R9)

//.breakpoint

LD(BP, -12, p\_array)

LD(BP, -16, p\_left)

LD(BP, -20, p\_right)

ADD(p\_left, p\_right, p\_pivotIndex)

SHRC(p\_pivotIndex, 1, p\_pivotIndex)

MULC(p\_pivotIndex, 4, R9)

ADD(p\_array, R9, R9)

LD(R9, 0, p\_pivotValue)

MULC(p\_right, 4, R9)

ADD(p\_array, R9, R9)

LD(R9, 0, R7)

MULC(p\_pivotIndex, 4, R9)

ADD(p\_array, R9, R9)

ST(R7, 0, R9)

MOVE(p\_left, p\_storeIndex)

START: ADDC(p\_left, 1, p\_left)

CMPLE(p\_left, p\_right, R8)

BF(R8, AFTER)

MULC(p\_left, 4, R9)

ADDC(R9, -4, R9)

ADD(p\_array, R9, R9)

LD(R9, 0, R8)

CMPLE(R8, p\_pivotValue, R1)

BF(R1, START)

MULC(p\_storeIndex, 4, R9)

ADD(p\_array, R9, R9)

LD(R9, 0, R1)

MULC(p\_left, 4, R9)

ADDC(R9, -4, R9)

ADD(p\_array, R9, R9)

ST(R1, 0, R9)

MULC(p\_storeIndex, 4, R9)

ADD(p\_array, R9, R9)

ST(R8, 0, R9)

ADDC(p\_storeIndex, 1, p\_storeIndex)

BR(START)

AFTER: MULC(p\_storeIndex, 4, R9)

ADD(p\_array, R9, R9)

LD(R9, 0, R8)

MULC(p\_right, 4, R9)

ADD(p\_array, R9, R9)

ST(R8, 0, R9)

MULC(p\_storeIndex, 4, R9)

ADD(p\_array, R9, R9)

ST(p\_pivotValue, 0, R9)

MOVE(p\_storeIndex, R0)

POP(R9)

POP(R8)

POP(R7)

POP(R6)

POP(R5)

POP(R4)

POP(R3)

POP(R2)

POP(R1)

MOVE(BP, SP)

POP(BP)

POP(LP)

JMP(LP)

//def quicksort(array, left, right):

// if left < right:

// pivotIndex = partition(array,left,right)

// quicksort(array,left,pivotIndex-1)

// quicksort(array,pivotIndex+1,right)

// quicksort(ArrayBase, left, right)

quicksort:

PUSH(LP)

PUSH(BP)

MOVE(SP, BP)

PUSH(R1)

PUSH(R2)

PUSH(R3)

PUSH(R4)

LD(BP, -12, R1)

LD(BP, -16, R2)

LD(BP, -20, R3)

.breakpoint

CMPLT(R2, R3, R4)

BF(R4, END)

PUSH(R3)

PUSH(R2)

PUSH(R1)

BR(partition, LP)

DEALLOCATE(3)

//.breakpoint

ADDC(R0, -1, R4)

PUSH(R4)

PUSH(R2)

PUSH(R1)

BR(quicksort, LP)

DEALLOCATE(3)

PUSH(R3)

ADDC(R0, 1, R4)

PUSH(R4)

PUSH(R1)

BR(quicksort, LP)

DEALLOCATE(3)

END: POP(R4)

POP(R3)

POP(R2)

POP(R1)

MOVE(BP, SP)

POP(BP)

POP(LP)

JMP(LP)