MIA: A 16-bit RISC ISA

Insn	Type	Operation
NOP	?	NOP
ADD	R	R[rd] <- R[rs] + R[rt]
ADDI	I	R[rs] <- R[rs] + sextimme
SUB	R	R[rd] <- R[rs] - R[rt]
AND	R	R[rd] <- R[rs] & R[rt]
OR	R	R[rd] <- R[rs] | R[rt]
SLT	R	R[rd] <- R[rs] < R[rt]
LW	R	R[rd] <- M[rt]
SW	R	M[rt] <- R[rs]
SWI	I	M[sextimme] <- R[rs]
BEZI	I	R[rs] == 0 ? PC <- sextimme
BNZI	I	R[rs] != 0 ? PC <- sextimme
BEZR	R	R[rs] == 0 ? PC <- R[rt]
BNZR	R	R[rs] != 0 ? PC <- R[rt]

R	op	rs	rt	rd
I	op	rs	imm	
	15 : 12	11 : 8	7 : 4	3 : 0

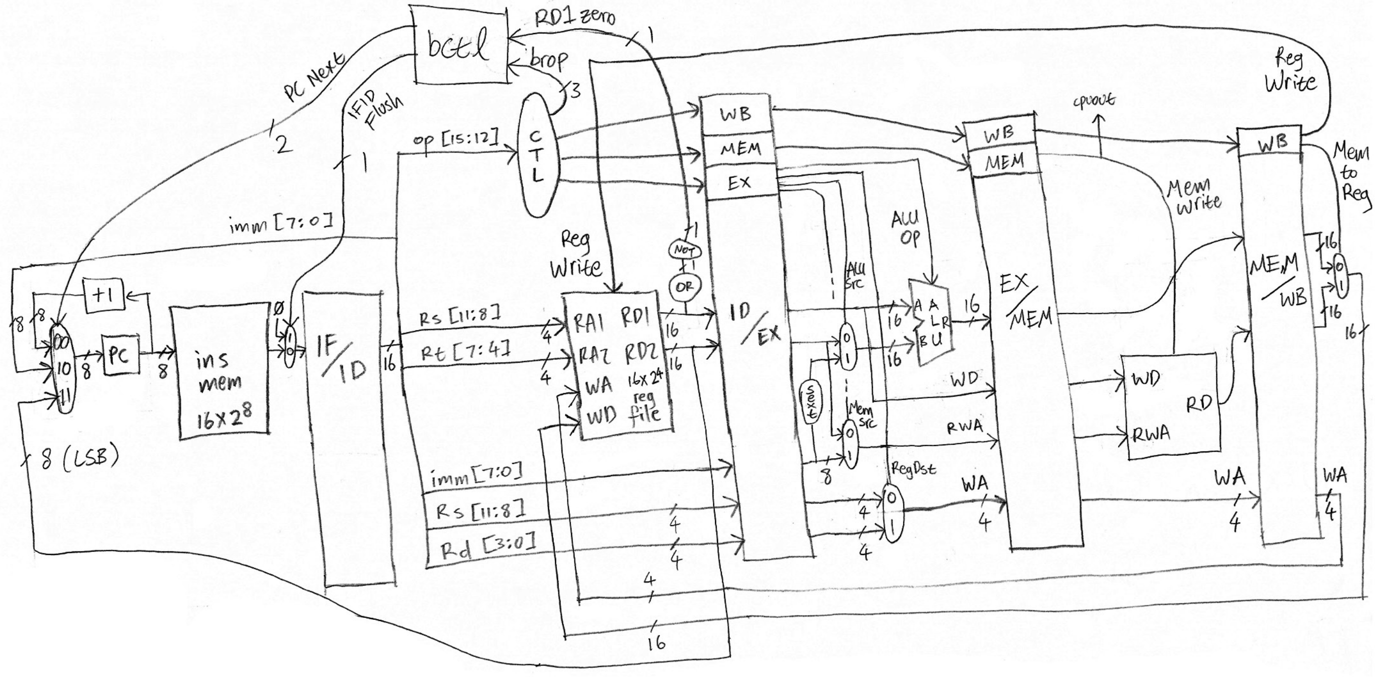
MIA Register Conventions

Reg	Usage
$0	Constant zero
$1	Function result
$2	Argument 1
$3	Argument 2
$4	Argument 3
$5	Argument 4
$6	Temporary
$7	Temporary
$8	Temporary
$9	Temporary
$10	Temporary
$11	Temporary
$12	Temporary
$13	Temporary
$14	Stack pointer
$15	Return address

MIA / ARCHEL Features

Pipelined, for extra efficiency
No compiler, for extra assembly fun
No assembler, for extra hand-linking fun
No offset immediate addressing,
for manual stack management fun

ARCHEL: A MIA Implementation

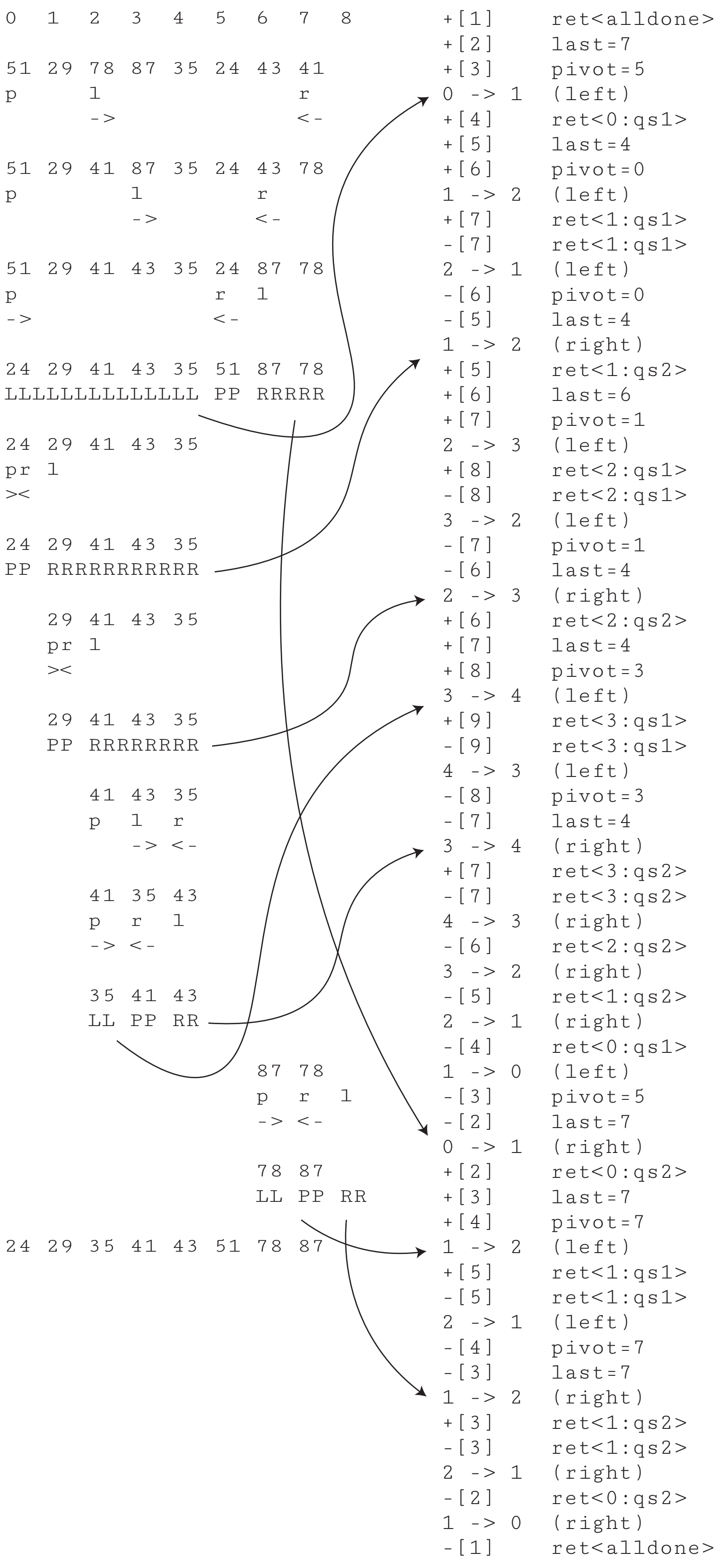


Quicksort in MIA

```
  0: addi $14 i+255
  1: addi $6 i+51
  2: addi $7 i+29
  3: addi $8 i+78
  4: addi $9 i+87
  5: addi $10 i+35
  6: addi $11 i+24
  7: addi $12 i+43
  8: addi $13 i+41
  9: swi $6 i+0
 10: swi $7 i+1
 11: swi $8 i+2
 12: swi $9 i+3
 13: swi $10 i+4
 14: swi $11 i+5
 15: swi $12 i+6
 16: swi $13 i+7
 17: addi $2 i+0
 18: addi $3 i+3
 19: addi $15 i+152
 20: nop
 21: nop
 22: nop
 23: nop
 24: quicksort:
 25: sw $15 $14 --
 26: addi $14 i-1
 27: slt $2 $3 $6
 28: nop
 29: nop
 30: bezi $6 :qs-end
 31: add $0 $0 $15
 32: nop
 33: nop
 34: addi $15 :qs0
 35: bezi $0 :partition
 36: qs0:
 37: sw $3 $14 --
 38: nop
 39: nop
 40: addi $14 i-1
 41: nop
 42: nop
 43: sw $1 $14 --
 44: nop
 45: nop
 46: addi $14 i-1
 47: add $1 $0 $3
 48: nop
 49: nop
 50: addi $3 i-1
 51: add $0 $0 $15
 52: nop
 53: nop
 54: addi $15 :qs1
 55: bezi $0 :quicksort
 56: qs1:
 57: addi $14 i+1
 58: nop
 59: nop
 60: lw -- $14 $2
 61: nop
 62: nop
 63: addi $2 i+1
 64: addi $14 i+1
 65: nop
 66: nop
 67: lw -- $14 $3
 68: add $0 $0 $15
 69: nop
 70: nop
 71: addi $15 :qs2
 72: bez $0 :quicksort
 73: qs2:
 74: qs-end:
 75: addi $14 i+1
 76: nop
 77: nop
 78: lw -- $14 $15
 79: nop
 80: nop
 81: nop
 82: bezr $0 $15 --
 83: nop
 84: nop
 85: nop
 86: nop
 87: partition:
 88: add $0 $0 $6
 89: nop
```

```
 90: nop
 91: lw -- $2 $6
 92: add $0 $2 $7
 93: add $0 $3 $8
 94: nop
 95: nop
 96: addi $7 i+1
 97: add $0 $0 $9
 98: notdone:
 99: leftmarking:
100: slt $8 $7 $11
101: lw -- $7 $10
102: nop
103: nop
104: slt $6 $10 $12
105: nop
106: nop
107: or $11 $12 $11
108: nop
109: nop
110: bnzi $11 :rightmarking
111: addi $7 i+1
112: bezi $0 :leftmarking
113: rightmarking:
114: slt $8 $7 $11
115: lw -- $8 $10
116: nop
117: nop
118: slt $10 $6 $12
119: nop
120: nop
121: or $11 $12 $11
122: nop
123: nop
124: bnzi $11 :checkdone
125: addi $8 i-1
126: bezi $0 :rightmarking
127: checkdone:
128: slt $8 $7 $9
129: nop
130: nop
131: bnzi $9 :done
132: lw -- $7 $10
133: lw -- $8 $11
134: nop
135: nop
136: sw $10 $8 --
137: sw $11 $7 --
138: bezi $0 :notdone
139: done:
140: lw -- $2 $10
141: lw -- $8 $11
142: nop
143: nop
144: sw $10 $8 --
145: sw $11 $2 --
146: add $0 $8 $1
147: bezr $0 $15 --
148: nop
149: nop
150: nop
151: nop
152: alldone:
153: add $0 $0 $6
154: add $0 $0 $7
155: add $0 $0 $8
156: add $0 $0 $9
157: add $0 $0 $10
158: add $0 $0 $11
159: add $0 $0 $12
160: add $0 $0 $13
161: addi $6 i+0
162: addi $7 i+1
163: addi $8 i+2
164: addi $9 i+3
165: addi $10 i+4
166: addi $11 i+5
167: addi $12 i+6
168: addi $13 i+7
169: lw -- $6 $6
170: lw -- $7 $7
171: lw -- $8 $8
172: lw -- $9 $9
173: lw -- $10 $10
174: lw -- $11 $11
175: lw -- $12 $12
176: lw -- $13 $13
177: halt:
178: bezi $0 :halt
```

Memory Trace



```
0  1  2  3  4  5  6  7  8

51 29 78 87 35 24 43 41
p     l              r
      ->             <-

51 29 41 87 35 24 43 78
p        l        r
         ->       <-

51 29 41 43 35 24 87 78
p              r  l
->             <-

24 29 41 43 35 51 87 78
LLLLLLLLLLLLLL PP RRRRR

24 29 41 43 35
pr l
><

24 29 41 43 35
PP RRRRRRRRRRR

   29 41 43 35
   pr l
   ><

   29 41 43 35
   PP RRRRRRRR

      41 43 35
      p  l  r
         -> <-

      41 35 43
      p  r  l
      -> <-

      35 41 43
      LL PP RR

                  87 78
                  p  r  l
                  -> <-

                  78 87
                  LL PP RR

24 29 35 41 43 51 78 87
```

Stack / Call Trace

```
+[1]     ret<alldone>
+[2]     last=7
+[3]     pivot=5
0 -> 1   (left)
+[4]     ret<0:qs1>
+[5]     last=4
+[6]     pivot=0
1 -> 2   (left)
+[7]     ret<1:qs1>
-[7]     ret<1:qs1>
2 -> 1   (left)
-[6]     pivot=0
-[5]     last=4
1 -> 2   (right)
+[5]     ret<1:qs2>
+[6]     last=6
+[7]     pivot=1
2 -> 3   (left)
+[8]     ret<2:qs1>
-[8]     ret<2:qs1>
3 -> 2   (left)
-[7]     pivot=1
-[6]     last=4
2 -> 3   (right)
+[6]     ret<2:qs2>
+[7]     last=4
+[8]     pivot=3
3 -> 4   (left)
+[9]     ret<3:qs1>
-[9]     ret<3:qs1>
4 -> 3   (left)
-[8]     pivot=3
-[7]     last=4
3 -> 4   (right)
+[7]     ret<3:qs2>
-[7]     ret<3:qs2>
4 -> 3   (right)
-[6]     ret<2:qs2>
3 -> 2   (right)
-[5]     ret<1:qs2>
2 -> 1   (right)
-[4]     ret<0:qs1>
1 -> 0   (left)
-[3]     pivot=5
-[2]     last=7
0 -> 1   (right)
+[2]     ret<0:qs2>
+[3]     last=7
+[4]     pivot=7
1 -> 2   (left)
+[5]     ret<1:qs1>
-[5]     ret<1:qs1>
2 -> 1   (left)
-[4]     pivot=7
-[3]     last=7
1 -> 2   (right)
+[3]     ret<1:qs2>
-[3]     ret<1:qs2>
2 -> 1   (right)
-[2]     ret<0:qs2>
1 -> 0   (right)
-[1]     ret<alldone>
```