

Reference Program – Testing and Verification

```

11
12 init:      # function to initialise PDF buffer memory
13 | LI      a1, 0x100      # loop_count a1 = 256
14 | _loop1:      # repeat
15 | ADDI    a1, a1, -1      # decrement a1
16 | SB      zero, base_pdf(a1) # mem[base_pdf+a1] = 0
17 | BNE     a1, zero, _loop1 # until a1 = 0
18 | RET

```

Figure 1.1 – initialise PDF array

```

a1 = 0x100
while a1 != 0:
    a1 -= 1
    mem[base_pdf+a1] = 0

```

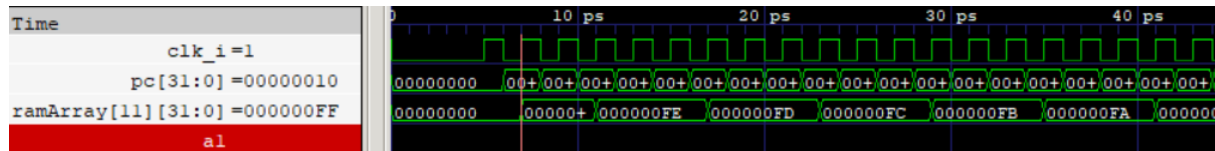


Figure 1.2 – function start

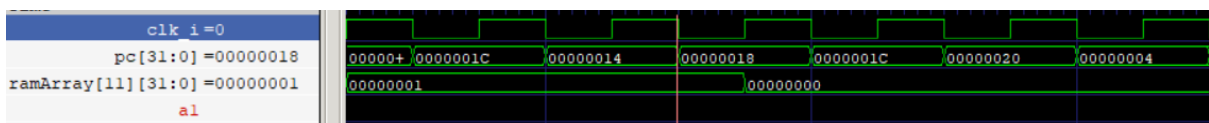


Figure 1.3 – function end

Success Criteria:

- JAL executing correctly – init function jumped to by main as seen by initial PC value. (Fig1.2)
- Data Memory mapping between a1 and register x11 (ramArray[11])
- Successful initialisation of a1 = 0x100 by LI and subsequent decrementing by 1 by ADDI by BNE correctly evaluating a1!=0 and executing branch.
- BNE passed at end of loop and successful Jump back to return address seen (Fig1.3)

```

build:      # function to build prob dist func (pdf)
LI      a1, base_data      # a1 = base address of data array
LI      a2, 0              # a2 = offset into of data array
LI      a3, base_pdf       # a3 = base address of pdf array
LI      a4, max_count      # a4 = maximum count to terminate
_loop2:      # repeat
ADD     a5, a1, a2          # a5 = data base address + offset
LBU     t0, 0(a5)          # t0 = data value
ADD     a6, t0, a3          # a6 = index into pdf array
LBU     t1, 0(a6)          # t1 = current bin count
ADDI    t1, t1, 1          # increment bin count
SB      t1, 0(a6)          # update bin count
ADDI    a2, a2, 1          # point to next data in array
BNE     t1, a4, _loop2     # until bin count reaches max
RET

```

Figure 2 – build PDF

```

_loop2:
while t1 != a4:
    a5 = a1 + a2
    t0 = mem[a5]
    a6 = t0 + a3
    t1 = mem[a6]
    t1 += 1
    mem[a6] = t1
    a2 += 1

```

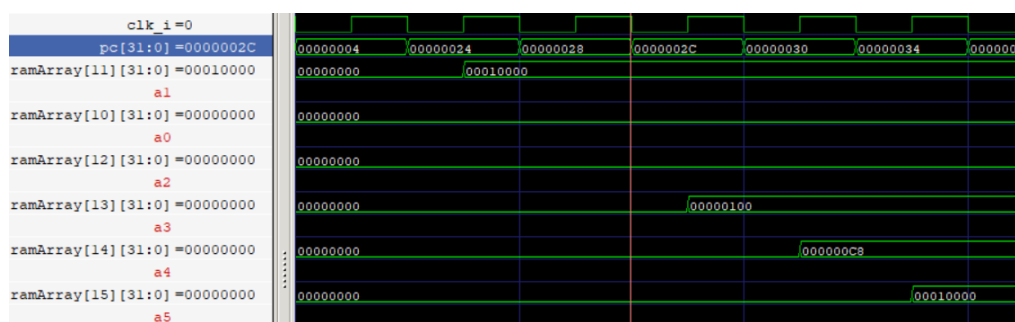


Figure 2.1 – initial variable storage

Reference Program – Testing and Verification

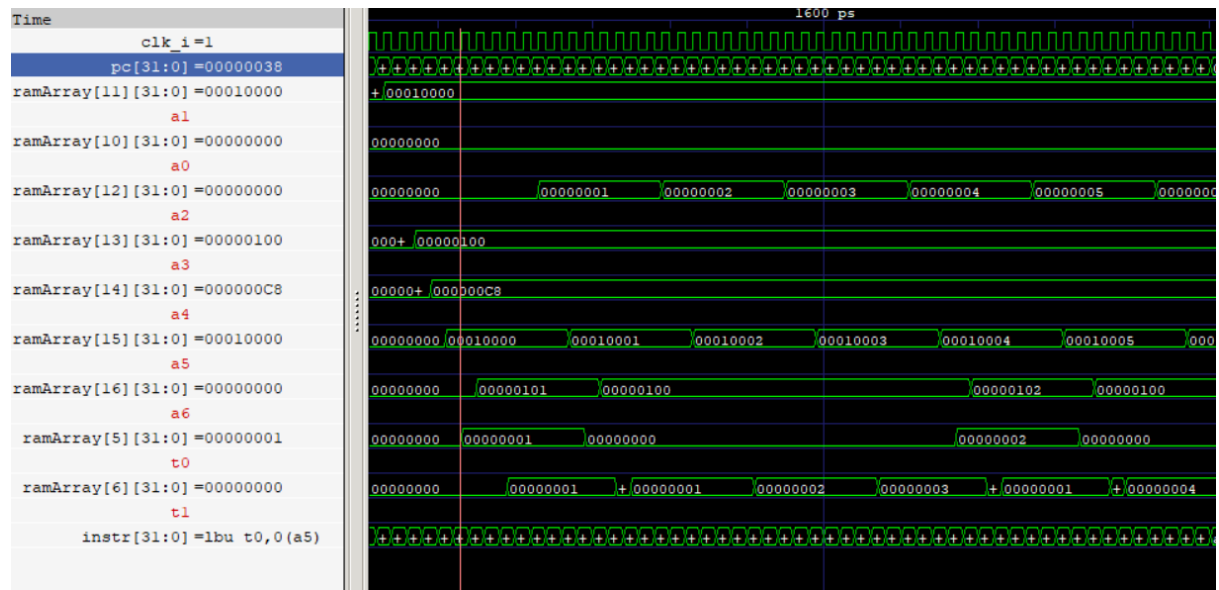


Figure 2.2 – `_loop2` for data array “triangle.mem”

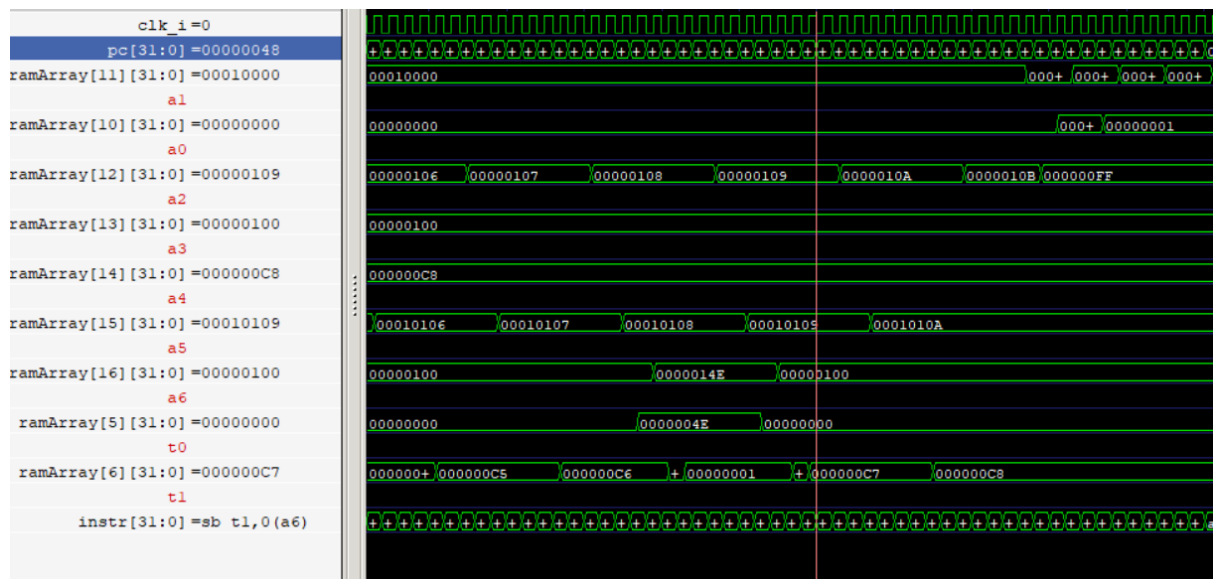


Figure 2.3 – `_loop2` end sequence

Success Criteria:

- Initial data values from triangle.mem 0x1 and 0x2 being read into t0 (Fig2.2)
- Data value being offset by 0x100 to get index in pdf_array, storing this into a6. 0x101 and 0x102 for the segment shown above.
- End of loop being successfully reached (PDF array build finished) and return to main to begin display function as bin count in t1 found to equal max_count (Fig2.3)