Assembly Program Analysis

Simulation Result

The table below follows the full code provided in this document, table below shows the effect of `JMP` and `PRESENT` as well.

Some degree of pipelining are applied to instructions that involves load and storage, the result will show up in the next cycle (next `IF1`).

Code	Time Duration	Men\Reg\Port Location
`NOOP`	$0ns - 160ns$	n\a
`LDR R1 #123`	$160ns - 400ns$	`R1`
`AND R2 R1 #111`	$400ns - 640ns$	`R2`
`LDR R3 R2`	$640ns - 840ns$	`R3`
`LDR R4 $2`	$840ns - 1120ns$	`R4`
`SUBV R4 R1 #23`	$1120ns - 1360ns$	`R4`
`CLFZ`	$1120ns - 1520ns$	n\a
`SUB R1 #23`	$1520ns - 1760ns$	n\a
`STR R2 $100`	$1760ns - 2000ns$	`$100`
`LDR R6 $100`	$2000ns - 2280ns$	`R6`
`ADD R7 R7 R3`	$2280ns - 2440ns$	`R7`
`PRESENT R8 $16`	$2440ns - 2680ns$	n\a
`NOOP`	$2680ns - 2840ns$	n\a
`NOOP`	$2840ns - 3000ns$	n\a
`NOOP`	$3000ns - 3160ns$	n\a
`SSOP R1`	$3160ns - 3320ns$	`SOP`
`LSIP R11`	$3320ns - 3480ns$	`R11`
`MAX R1 #200`	$3480ns - 3720ns$	`R1`
`DCALLBL R2 #123`	$3720ns - 4840ns$ (blocked)	`R0`

`AND R9 R1 #111`	$4840ns - 5080ns$	`R9`
`DCALLNB R2 #100`	$5080ns - 5320ns$	`DPCR`
`ADD R14 R13 #10`	$5320ns - 5560ns$	`R14`
`LDR R14 $0`	$5560ns - 5920ns$	`R14`
`NOOP`	$5920ns - 6080ns$	n\a
`NOOP`	$6080ns - 6240ns$	n\a
`LDR R14`	$6240ns - 6520ns$	`R14`
`SUB R14 #3`	$6520ns - 6760ns$	n\a
`OR R15 R14 #22`	$6760ns - 7000ns$	`R15`
`OR R15 R15 R3`	$7000ns - 7160ns$	`R15`
`STR R12 #55`	$7160ns - 7400ns$	`$0`
`LDR R2 $0`	$7400ns - 7680ns$	`R2`
`STR R12 R3`	$7680ns - 7840ns$	`$0`
`LDR R2 $0`	$7840ns - 8120ns$	`R2`
`SEOT`	$8120ns - 8280ns$	`EOT`
`SSVOP R4`	$8280ns - 8440ns$	`SVOP`
`CEOT`	$8440ns - 8600ns$	`EOT`
`LER R0`	$8600ns - 8760ns$	`R0`
`CER`	$8760ns - 8920ns$	`ER`
`STRPC $0`	$8920ns - 9160ns$	`$0`
`LDR R2 $0`	$9160ns - 9440ns$	`R2`
`AND R0 R8 #20`	$9440ns - 9680ns$	`R0`
`JMP 20`	$9680ns - 9920ns$	`$20`
`CLFZ`	$9920ns - 10120ns$	`Z`
`NOOP`	$10120ns - 10240ns$	n\a
`NOOP`	$10240ns - 10400ns$	n\a
`NOOP`	$10400ns - 10560ns$	n\a
`NOOP`	$10560ns - 10720ns$	n\a

`SSOP R1`	$10720ns - 10880ns$	`SOP`
`LSIP R11`	$10880ns - 11040ns$	`R11`
`MAX R1 #200`	$11040ns - 11280ns$	`R1`
`DCALLNB R2 #100`	$11280ns - \infty$ (blocked)	`DPCR`

Full Code Used for testing ReCOP

```
start NOOP
  LDR R1 #123
  AND R2 R1 #111
  LDR R3 R2
  LDR R4 $2
  SUBV R4 R1 #23
  CLFZ
  SUB R1 #23

  STR R2 $100
  LDR R6 $100
  ADD R7 R7 R3
  PRESENT R8 $16
  CLFZ
  NOOP
  NOOP
  NOOP
  NOOP
  SSOP R1
  LSIP R11
  MAX R1 #200
  DCALLBL R2 #123
  AND R9 R1 #111
  DCALLNB R2 #100
  ADD R14 R13 #10
  LDR R14 $0
  NOOP
  NOOP
  LDR R14 $0
  SUB R14 #3
  OR R15 R14 #22
  OR R15 R15 R3
  STR R12 #55
  LDR R2 $0
  STR R12 R3
  LDR R2 $0
  SEOT
  SSVOP R4
  CEOT
  LER R0
  CER
  STRPC $0
  LDR R2 $0
  AND R0 R8 #20
  JMP 20
  NOOP
  DCALLBL R0

ENDPROG
```