

The diagram illustrates the data paths between Memory and Processor:

- Memory:** A large block on the left. It has a curved arrow indicating a range of 2^{12} locations. It is connected to the Processor via a 32-bit Address bus (pointing to Memory) and a 32-bit Data bus (bidirectional).
- Processor:** A large block on the right. It contains:
 - PC (Program Counter):** 12 bits.
 - IR (Instruction Register):** 32 bits.
 - PSR (Program Status Register):** 5 bits.
 - Register File:** Connected to the ALU via a 2^{12} bit bus.
 - ALU (Arithmetic Logic Unit):** A large trapezoidal block.
- CLOCK:** A box at the top left with an arrow pointing to the Processor.

Msgs											
/nplCPU/lr	xxx...			2800 100 1			3800 10 10			200 10000	
/nplCPU/src1	xxx...										
/nplCPU/src2	xxx...										
/nplCPU/result	xxx...										
/nplCPU/psr	xxx										
/nplCPU/pc	xxx		00		04						
/nplCPU/pc		000		001			002			003	
/nplCPU/dir	x										
/nplCPU/reset	1										
/nplCPU/i	xxx...		000000 10								

```
# Time=0 pc= x lr=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx r0=xxxxxxxx r1=xxxxxxxx r2=xxxxxxxx MEM16=xxxxxxxx psr=xxxxxx
# Time=10000 pc= 0 lr=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx r0=00000000 r1=00000000 r2=00000000 MEM16=xxxxxxxx psr=000000
# Time=20000 pc= 1 lr=00101000000000000000001000000000000001 r0=00000000 r1=00000000 r2=00000000 MEM16=xxxxxxxx psr=000000
# Time=30000 pc= 1 lr=00101000000000000000001000000000000001 r0=00000000 r1=00000001 r2=00000000 MEM16=xxxxxxxx psr=001000
# Time=50000 pc= 2 lr=001110000000000000000010000000100000 r0=00000000 r1=00000001 r2=00000000 MEM16=xxxxxxxx psr=001000
# Time=60000 pc= 2 lr=001110000000000000000010000000100000 r0=00000000 r1=00000001 r2=00000000 MEM16=00000001 psr=001000
# Time=80000 pc= 3 lr=001000000000000000000010000000000000000000 r0=00000000 r1=00000001 r2=00000000 MEM16=00000001 psr=001000
# Time=90000 pc= 3 lr=001000000000000000000010000000000000000000 r0=00000001 r1=00000001 r2=00000000 MEM16=00000001 psr=001000
```

指令	影響	說明
0: LD R1, #1	$R1 = 1$	讀取 1 放進 reg1
1: STR MEM[16], R1	$MEM[16] = R1$	把 reg1 的值放到 MEM[16]
2: LD R0, MEM[16]	$R0 = MEM[16]$	讀取 MEM[16] 的值放進 reg1
3: ADD R1, #1	$R1 = R1 + 1 = 2$	把 reg1 的值加 1

	Msgs								
/nplCPU/ir	600...	48001001		28002002		50002001		a8005000	
/nplCPU/src1	000...	00000001				00000002		00000005	
/nplCPU/src2	000...	00000001				00000002		00000001	
/nplCPU/result	1fffff...	00000002				00000004		00000005	
/nplCPU/psr	13	04	05					00	
/nplCPU/pc	008	004		005		006		007	
/nplCPU/dir	x								
/nplCPU/reset	0								
/nplCPU/i	000...	00000010							

```
# Time=110000 pc= 4 ir=01001000000000000001000000000001 r0=00000001 r1=00000001 r2=00000000 MEM16=00000001 psr=00100
# Time=120000 pc= 4 ir=01001000000000000001000000000001 r0=00000001 r1=00000001 r2=00000000 MEM16=00000001 psr=00110
# Time=130000 pc= 4 ir=01001000000000000001000000000001 r0=00000001 r1=00000002 r2=00000000 MEM16=00000001 psr=00110
# Time=140000 pc= 5 ir=001010000000000000010000000000010 r0=00000001 r1=00000002 r2=00000000 MEM16=00000001 psr=00110
# Time=150000 pc= 5 ir=001010000000000000010000000000010 r0=00000001 r1=00000002 r2=00000002 MEM16=00000001 psr=00110
# Time=170000 pc= 6 ir=010100000000000000010000000000001 r0=00000001 r1=00000002 r2=00000002 MEM16=00000001 psr=00110
# Time=190000 pc= 6 ir=010100000000000000010000000000001 r0=00000001 r1=00000004 r2=00000002 MEM16=00000001 psr=00110
# Time=200000 pc= 7 ir=1010100000000000000101000000000000 r0=00000001 r1=00000004 r2=00000002 MEM16=00000001 psr=00110
# Time=210000 pc= 7 ir=1010100000000000000101000000000000 r0=00000001 r1=00000004 r2=00000002 MEM16=00000001 psr=00000
# Time=220000 pc= 7 ir=1010100000000000000101000000000000 r0=00000005 r1=00000004 r2=00000002 MEM16=00000001 psr=00000
```

指令	影響	說明
4: LD R2, #2	R2 = 2	讀取 2 放進 reg2
5: MUL R1, R2	R1 = R1 * R2 = 4	把 reg1 的值與 reg2 的值相乘放到 reg1
6: OR R0, #5	R0 = R0 OR 5	將 reg0 的值與 5 進行 OR 運算
7: CMP R2	R2 = 2 的補數	取 reg2 值的 2 的補數

	Msgs								
/nplCPU/ir	780...	60000002		88002000		88ffe000		b0001002	
/nplCPU/src1	000...	00000005		00000002		000000fe		00000004	
/nplCPU/src2	fffffffa	00000001		00000005		40000001		fffffffa	
/nplCPU/result	000...	000000005	1fffffffa	040000001		000000005		000000000	
/nplCPU/psr	0a	00	13	00				0a	
/nplCPU/pc	00c	008		009		00a		00b	
/nplCPU/dir	0								
/nplCPU/reset	0								
/nplCPU/i	000...	00000010		00000000					

```
# Time=230000 pc= 8 ir=01100000000000000000000000000001 r0=00000005 r1=00000004 r2=00000002 MEM16=00000001 psr=00000
# Time=240000 pc= 8 ir=01100000000000000000000000000010 r0=00000005 r1=00000004 r2=00000002 MEM16=00000001 psr=10011
# Time=250000 pc= 8 ir=01100000000000000000000000000010 r0=00000005 r1=00000004 r2=fffffffa MEM16=00000001 psr=10011
# Time=260000 pc= 9 ir=1000100000000000000100000000000000 r0=00000005 r1=00000004 r2=fffffffa MEM16=00000001 psr=10011
# Time=270000 pc= 9 ir=1000100000000000000100000000000000 r0=00000005 r1=00000004 r2=fffffffa MEM16=00000001 psr=00000
# Time=280000 pc= 9 ir=1000100000000000000100000000000000 r0=40000001 r1=00000004 r2=fffffffa MEM16=00000001 psr=00000
# Time=290000 pc= 10 ir=10001000111111111100000000000000 r0=40000001 r1=00000004 r2=fffffffa MEM16=00000001 psr=00000
# Time=310000 pc= 10 ir=10001000111111111100000000000000 r0=00000005 r1=00000004 r2=fffffffa MEM16=00000001 psr=00000
# Time=320000 pc= 11 ir=101100000000000000010000000000010 r0=00000005 r1=00000004 r2=fffffffa MEM16=00000001 psr=00000
# Time=330000 pc= 11 ir=101100000000000000010000000000010 r0=00000005 r1=00000004 r2=fffffffa MEM16=00000001 psr=01010
# Time=340000 pc= 11 ir=101100000000000000010000000000010 r0=00000005 r1=00000004 r2=00000000 MEM16=00000001 psr=01010
```

指令	影響	說明
8: ROT R0, #2	右旋轉 R0 兩位	將數據位元循環右移 2 位
9: ROT R0, #-2	左旋轉 R0 兩位	將數據位元循環左移 2 位
10: AND R2, R1	R2 = R2 & R1	將 reg1 的值與 reg1 的值進行 AND 運算
11: SHF R0, #1	右移 R0 一位	取 reg0 值除 2

```
# Time=350000 pc= 12 ir=0111110000000000000010000000000000 r0=00000005 r1=00000004 r2=00000000 MEM16=00000001 psr=01010
# Time=360000 pc= 12 ir=011111000000000000000010000000000000 r0=00000005 r1=00000004 r2=00000000 MEM16=00000001 psr=00110
# Time=370000 pc= 12 ir=011111000000000000000010000000000000 r0=00000002 r1=00000004 r2=00000000 MEM16=00000001 psr=00110
# Time=380000 pc= 13 ir=01111100011111111111110000000000000000 r0=00000002 r1=00000004 r2=00000000 MEM16=00000001 psr=00110
# Time=400000 pc= 13 ir=01111100011111111111110000000000000000 r0=00000004 r1=00000004 r2=00000000 MEM16=00000001 psr=00110
# Time=410000 pc= 14 ir=00010010000000000000000000000000011111 r0=00000004 r1=00000004 r2=00000000 MEM16=00000001 psr=00110
# Time=420000 pc= 15 ir=00010010000000000000000000000000001111 r0=00000004 r1=00000004 r2=00000000 MEM16=00000001 psr=00110
# Time=440000 pc= 16 ir=1001000000000000000000000000000000000000 r0=00000004 r1=00000004 r2=00000000 MEM16=00000001 psr=00110
# Halt instruction executed at time 450000
# ** Note: $stop : C:/NCKU/VLSI/HW5_3/nplCPU.v(242)
# Time: 450 ns Iteration: 0 Instance: nplCPU
```

指令	影響	說明
12: SHF R0, #-1	左移 R0 一位	取 reg0 值乘 2
13: BRA 15, EVEN	如果 EVEN=1 則跳到 HALT	
14: ADD R1, #1	如果不是偶數，R1 加 1	不執行
15: HALT	停止	