Midterm Solution

problem 1.

1. Deadlock：資源互相依賴，造成pkt無法順利抵達Dst.
   可用turn model來限制，解決此問題

   Livelock：pkt不斷在Network上走，但一直無法抵達Dst.
   可用routing方向限制來解決

   starvation：某方向進來的pkt，一直無法被service到.
   可用abiter設計來解決

2. Deterministic：在runtime过程中，routing path固定.
   pro：設計簡單
   con：容易造成traffic conjection.

   Adaptive：依據当下網路狀況，来選擇較合適的path
   pro：path diversity大，所以不容易有traffic Conjection
   con：設計較複雜

3. Source-table routing：所有pkt在由source送出時，即決定path，並記錄在header中.
   pro：可optimize traffic　con：header size大

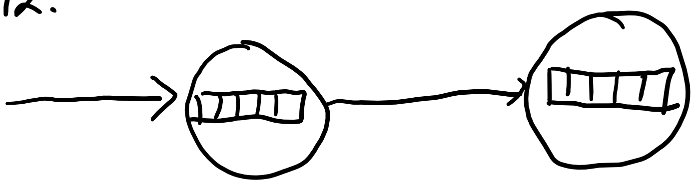
   Node-table routing：所有pkt在行經每个node時，透过查表來決定要往何处去。
   pro：routing Computing unit小
   con：每个node都有table，area overhead大

Algorithmic routing: pkt行經node時在進行routing path的運算

pro: node area size 較小

con: computing latency 長.

Problem2.



1、Assume one pkt exist in the current router

5 cycles to deliver Pkt-1 to next router

5 cycles to store Pkt-2 to the current router.

5 cycles to forward Pkt-2 to the next router.

∴ 5+5+5=15 cycles

Sol2. If pkt-1 comes from the previous upstream router. the solution needs to consider the latency to deliver Pkt-1 (i.e., 5 cycles)

→ Answer is 20 cycles

2、Assume two pkts exist in the two virtual channels of current router,

5 cycles to deliver Pkt-1 to next router (winner take all)

5 cycles to deliver Pkt-2 to next router

∴ 5+5=10 cycles

Pkt-1 come from the other node

Sol2. If Pkt-1 & Pkt-2 come from the other router. answer is 5+5+5=15 cycles

pk1 to next node

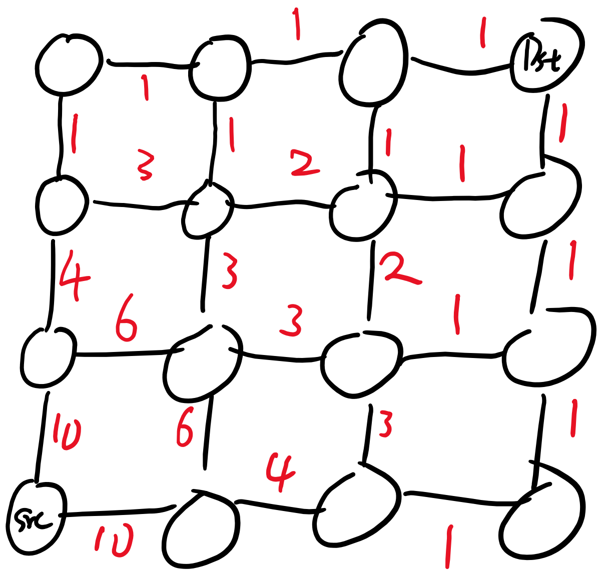
pk2 to current node.

3. 2 Pkts ⇒ 10 flits

∴ 10 cycles.

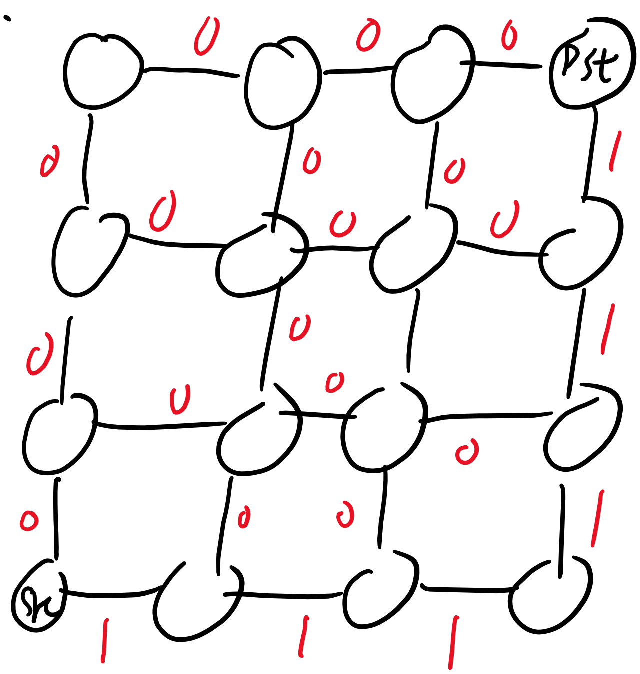
Problem 3.

1.



Ans: 20 paths

2.



Ans: 1 path

3.

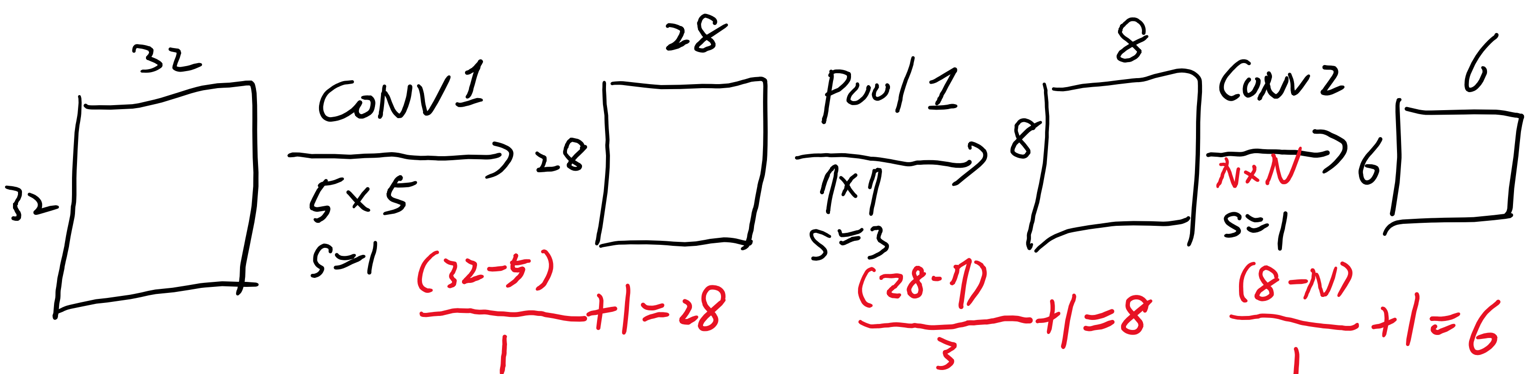
	1	2	3	4	5	6	7	8	9	10	11	12...
Head	RC	VA	SA	ST	RC	VA	SA	ST				
B1				SA	ST			SA	ST			
B2					SA	ST			SA	ST		
B3						SA	ST			SA	ST	
Tail							SA	ST			SA	ST

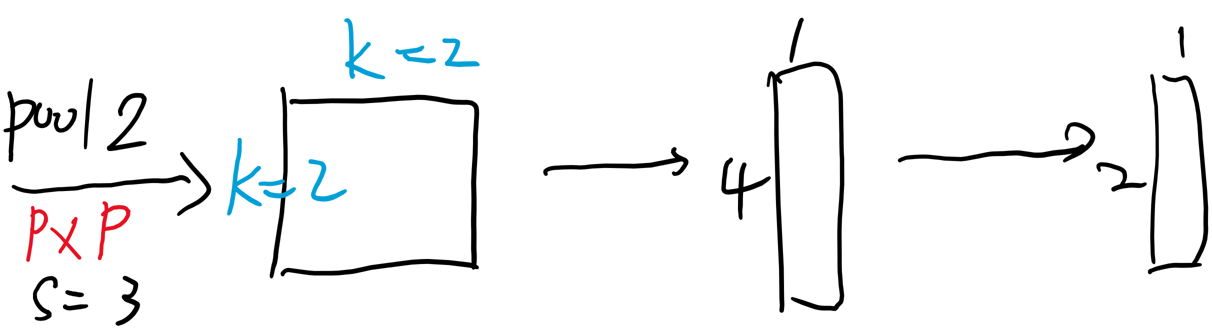
next hop

∵ 8 cycles for the first hop ; 4 cycles for the remaining hops

∴ $\underbrace{8+4+4+4+4+4}_{6 \text{ hops}} = 28$ cycles

Problem4





1、 ∵ output map after the pool2 is square

∴ $k=2$

∵ $\frac{(8-N)}{1}+1=6$ ∴ $N=3$

∵ $\frac{(6-P)}{3}+1=2$ ∴ $P=3$

2、 Parameters.

CONV1 : $5\times5\times1\times1=25$ (Assume channel#=1. M=1)

POOL1 : 0

CONV2 : $3\times3\times1\times1=9$

pool2 : 0

FC1 : $4\times4=16$

FC2 : $4\times2=8$

∴ $25+9+16+8=58$

Consider biases

sol2. $(25+\underline{1})+(9+\underline{1})+(4\times4+\underline{1\times4})+(4\times2+\underline{1\times2})=66$

Operations.

CONV1: $5\times5\times28\times28 = 19600$

pool1: $8\times8\times7\times7 = 3136$

CONV2: $3\times3\times6\times6 = 324$

pool2: $2\times2\times3\times3 = 36$

FC1: $4\times4 = 16$

FC2: $4\times2 = 8$

$\therefore 19600+3136+324+36+16+8 = 23120$

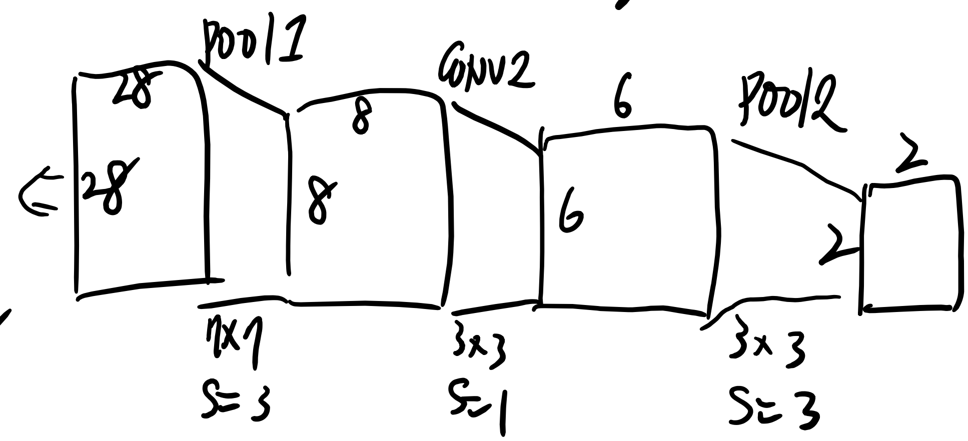
3. The map size after the second pool layer is 2×2

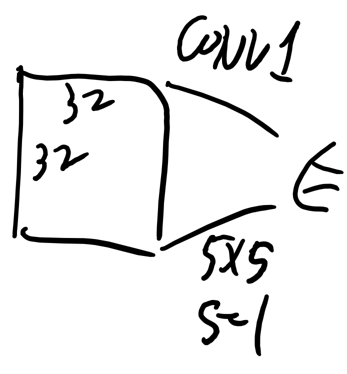
$(2-1)\times3+3=6$

$(6-1)\times1+3=8$

$(8-1)\times3+7=28$

$(28-1)\times1+5=\underline{\underline{32}}$





$\therefore$ It can cover all input data problem5.

1. OS: keep partial sums cached on PEs - work on subset of output at a time

WS: keep weights cached in PE register files. Each activation is broadcast to all PEs, and computed partial sum is forwarded to other PEs to complete computation.

IS: Unicast weights and accumulate partial sums spatially across the PE array.

RS: keep as much related to the same filter row cached across PEs

2.

$$OS: \underbrace{E\times R}_{\text{weight}} + \underbrace{E\times R}_{\text{input}} + \underbrace{0}_{\text{output}}$$

$$WS: \underbrace{R}_{\text{weight}} + \underbrace{E\times R}_{\text{input}} + \underbrace{E\times R}_{\text{output}}$$

$$IS: \underbrace{E\times R}_{\text{weight}} + \underbrace{H}_{\text{input}} + \underbrace{E\times R}_{\text{output}}$$

$$RS: R + H + 0$$

	E	R	H
CONV1	28x28	5x5	32x32
CONV2	6x6	3x3	8x8

CONV1: $OS = (28\times28\times5\times5) + (28\times28\times5\times5) + 0 = 39200$

$WS = (5\times5) + (28\times28\times5\times5) + (28\times28\times5\times5) = 39225$

$IS = (28\times28\times5\times5) + (32\times32) + (28\times28\times5\times5) = 40224$

$RS = (5\times5) + (32\times32) + 0 = 1049$

CONV2: $OS = (6\times6\times3\times3) + (6\times6\times3\times3) + 0 = 648$

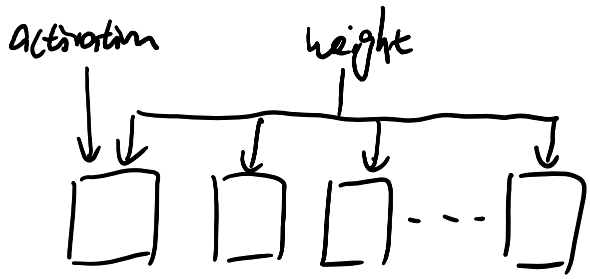
$WS = (3\times3) + (6\times6\times3\times3) + (6\times6\times3\times3) = 657$

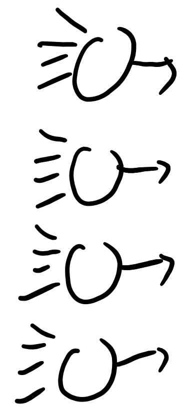
$IS = (6\times6\times3\times3) + (8\times8) + (6\times6\times3\times3) = 712$

$RS = (3\times3) + (8\times8) + 0 = 73$

FC1: OS

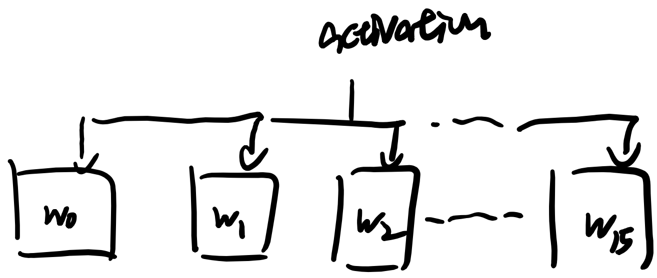
(no output read)





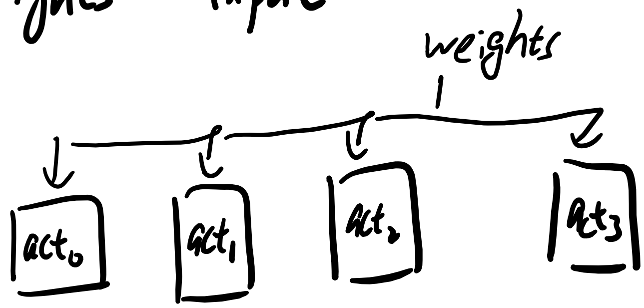
$\therefore \underbrace{4\times4}_{\text{weights}} + \underbrace{4\times4}_{\text{input}} = 32$

WS



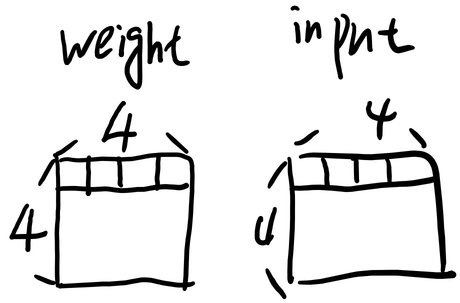
$\therefore \underbrace{16}_{\text{weights}} + \underbrace{4\times4}_{\text{input}} = 32$

IS



$\underbrace{4}_{\text{input}} + 4\times4 = 20$

RS



$\therefore 4\times4+4\times4$
$= 32$

FC2

$OS = 4\times2 + 4\times2 = 16$

$WS = 8 + 4\times2 = 16$

$IS = 4 + 4\times2 = 12$

$RS = 4\times2 + 4\times2 = 16$

$\therefore$

$OS = 39200+648+32+16 = 39896$

$WS = 39225+657+32+16 = 39930$

$IS = 40224+712+20+12 = 40968$

$RS = 1049+73+32+16 = 1170$

3.

	CONV	POOL	FC
OS	weight: broadcast input: multicast	Multicast	weight: unicast input: broadcast
WS	input: broadcast psum: unicast weight: X	multicast	weight: X input: multicast
IS	input: X weight: unicast psum: unicast	multicast	input: X weight: unicast
RS	input: multicast weight: multicast psum: unicast	multicast	input: broadcast weight: unicast