

NYCU-EE IC LAB – FALL 2023

Lab04 Exercise

Design: Siamese Neural Network

Data Preparation

1. Extract files from TA's directory:

```
% openssl des3 -d -k zk5ZaSbq+yU= -salt -in ~iclabTA01/Lab04.tar | tar xvf -
```

2. The extracted LAB directory contains:

- 00_TESTBED
- 01_RTL
- 02_SYN
- 03_GATE
- 09_SUBMIT

Design Description

The *Siamese neural network* is a type of neural network architecture designed for similarity learning and feature extraction tasks. It is called "Siamese" because the network consists of two identical subnetworks, known as twin networks, which share the same architecture and parameters. These twin networks are used to process two different input samples.

The primary application of *Siamese neural network* is in tasks that involve determining similarity or dissimilarity between two input samples. For example, Siamese networks are commonly used in tasks like face recognition, signature verification, one-shot learning, and similarity-based recommender systems.

In this lab, you are asked to design a *Siamese Neural Network* accelerator Fig 1. You have to take two Convolution Neural Networks (CNN) with identical structures and weights as sub-networks and concatenate them together to form a *Siamese Neural Network*. The CNN sub-network includes operations such as **convolution**, **max-pooling**, **fully connected**, and **normalization**. The features computed by the two sub-networks are passed through an activation function to produce image encodings, which are then compared using the L1 distance to calculate the similarity score.

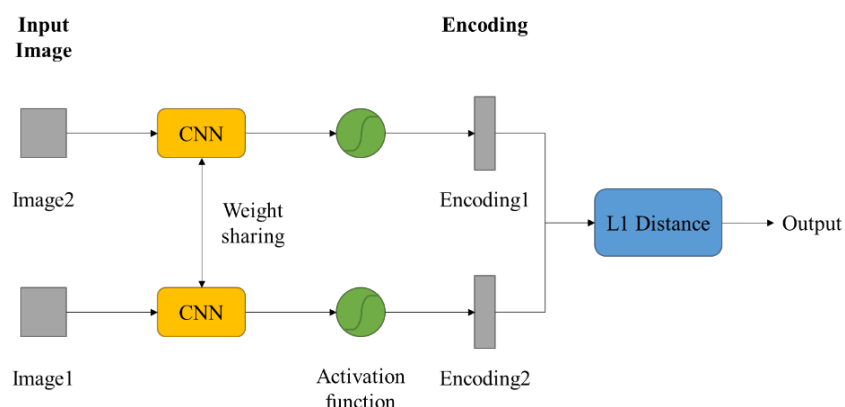


Fig 1. Siamese Neural Network architecture

- Description of input signals

When **in_valid** is high, the 32-bit **Img** signals will receive $4 \times 4 \times 3 \times 2 = 96$ cycles continuously to represent 2 input $4 \times 4 \times 3$ images.

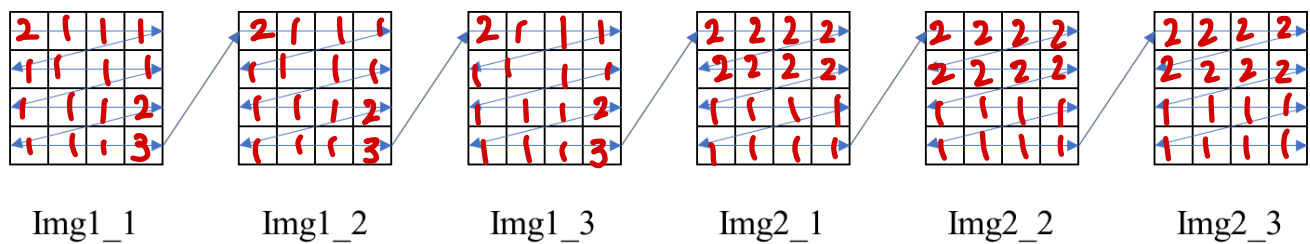


Fig 2. Sending order of the Img signal

The 32-bit unsigned **Kernel** signal will receive $3 \times 3 \times 3 = 27$ cycles continuously to represent the $3 \times 3 \times 3$ kernel.

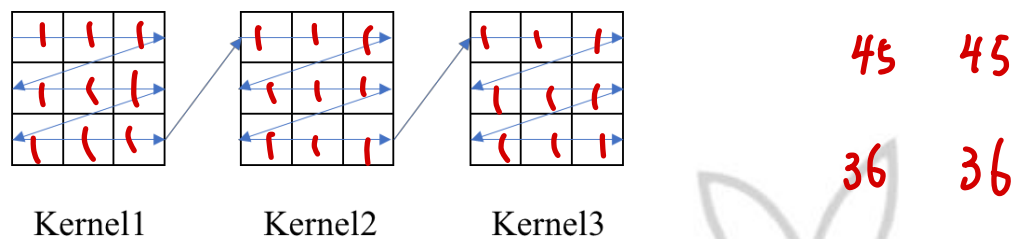


Fig 3. Sending order of the Kernel signal

The 32-bit **Weight** signal will also receive **4 cycles** continuously to represent the 2×2 matrix for the weight of the fully connected layer.

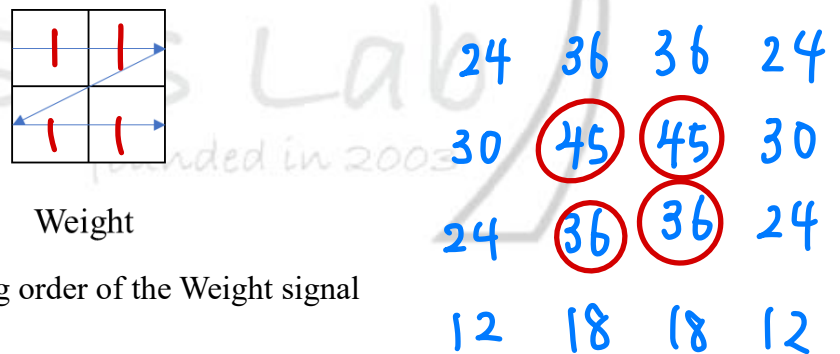
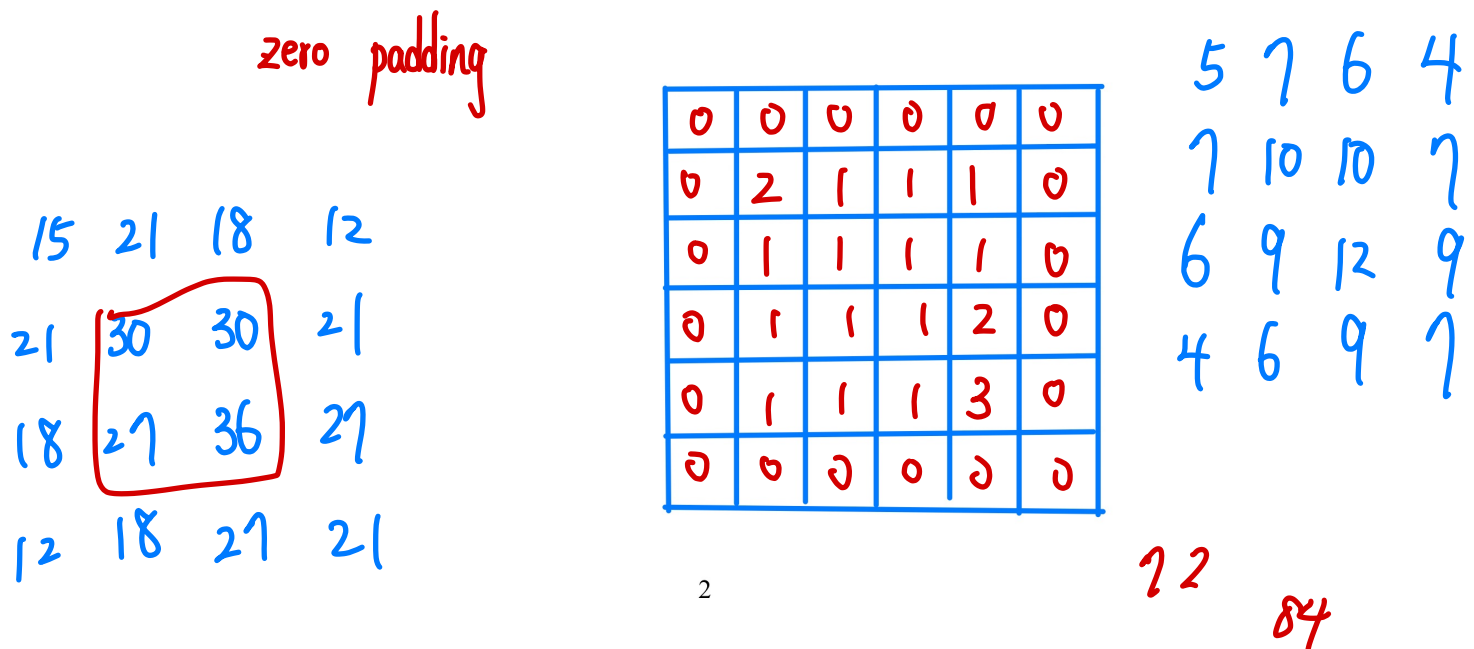


Fig 4. Sending order of the Weight signal

When the 2 images have been provided, i.e., after 96 cycles, **in_valid** will be pulled low. Note that the input signals **Img**, **Kernel**, and **Weight** are all sent in raster scan order.



39 33 1 1
27 57 1 1

- Description of CNN sub-networks

Before doing the convolution, you must perform **Replication Padding or Zero Padding** according to the information given by Opt. The padding width is 1.

The input of the CNN sub-network is a 32-bit 6x6x3 image. **Img1_1, Img1_2, Img1_3** will be fed into the upper sub-network, while **Img2_1, Img2_2, and Img2_3** will be fed into the lower sub-network. First, computing the convolution of the image with the kernel will result in a 4x4x1 feature map. Then, the 32-bit 4x4x1 feature map will do the max pooling operation, resulting in a 2x2x1 feature map. The next step is to input the 2x2 feature map to a fully connected layer, resulting in a 4x1 feature map. In this step, the 2x2 feature map will be multiplied by a 2x2 weight matrix, where **Weight** represents the matrix elements and will be continuously provided for 4 cycles during the input stage. The resulting matrix will be flattened into a 4x1 feature map. And then perform normalization.

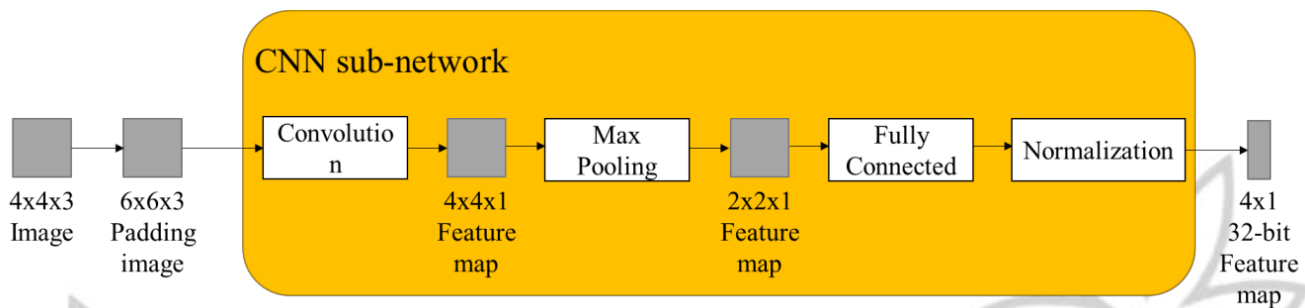


Fig 5. CNN sub-network

- Description of connection of two sub-networks

After computing the two sub-networks, the result is then passed through an **activation function**. The result of the activation function is called encoding vector. Next, the encoding vector passed through L1 distance calculation to obtain the **out** signal.

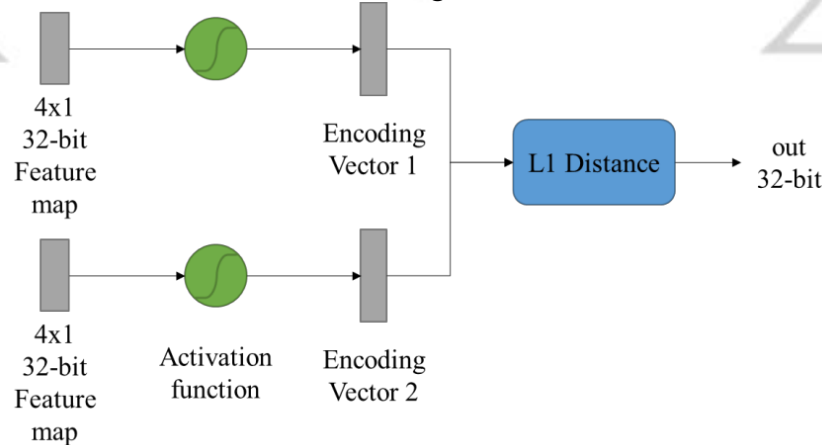


Fig6. Connection of two sub-network

- **Replication Padding**

Replication padding, is a technique used in image processing and computer vision to extend the borders of an image by replicating or mirroring the existing pixels.

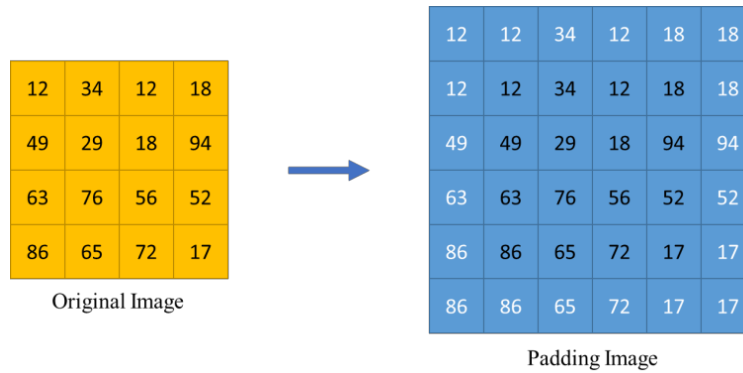


Fig7. Replication Padding

- **Zero Padding**

Zero padding involves adding zeros around the borders of an image or signal before applying certain operations, such as convolutions or Fourier transforms.

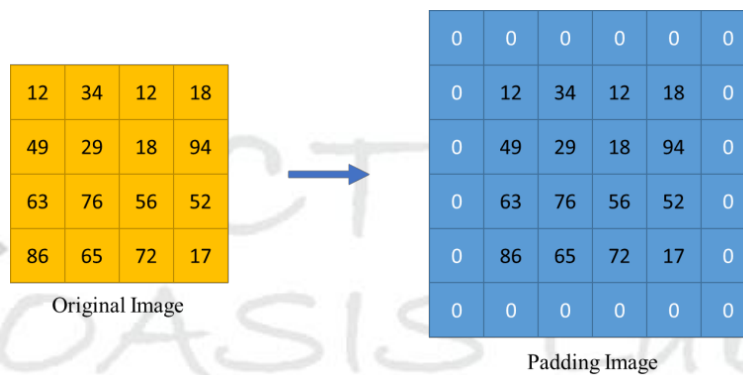


Fig8. Zero Padding

– Convolution

Formula of convolution:

$$FeatureMap[m,n] = \sum_j \sum_i Image[m,n] \cdot Kernel[m-i,n-j]$$

Ex:

$$5880 = 1*1+2*2+8*8+7*2+128*4+3*240+8*5+255*15+100*7$$

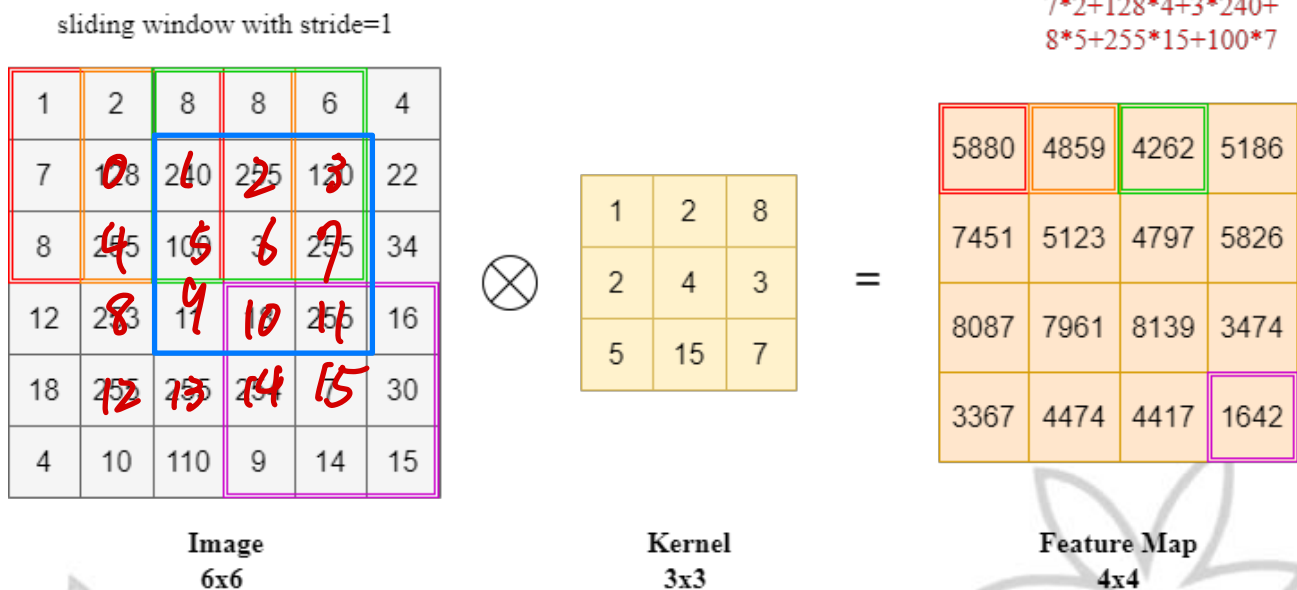


Fig9. Example of convolution operation

– Max-Pooling

The max-pooling operation works by sliding a 2x2 window, over the input feature map and taking the maximum value in each window as output.

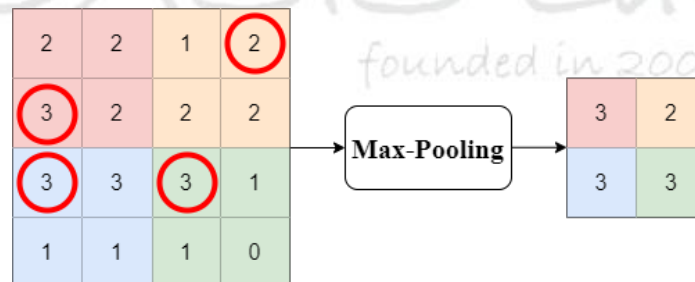


Fig10. Example of max-pooling operation

315
60

– Fully Connected

A fully connected layer can be represented as a matrix multiplication operation between the input matrix and weight matrix. Flattening the output matrix in obtaining a feature map.

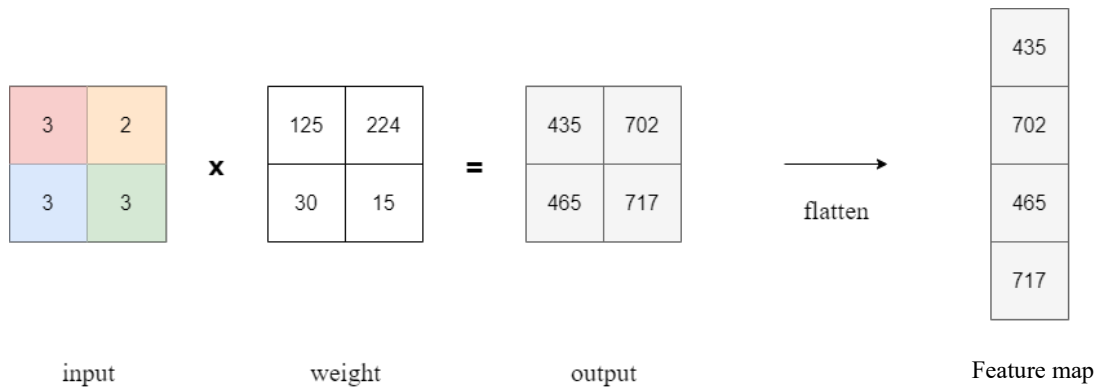


Fig11. Example of fully connected layer

– Min-Max Normalization

Min-Max Normalization is a data processing technique used to transform numeric features to a specific range, typically between 0 and 1.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

– Activation Function

An activation applies a non-linear transformation to the result of CNN.

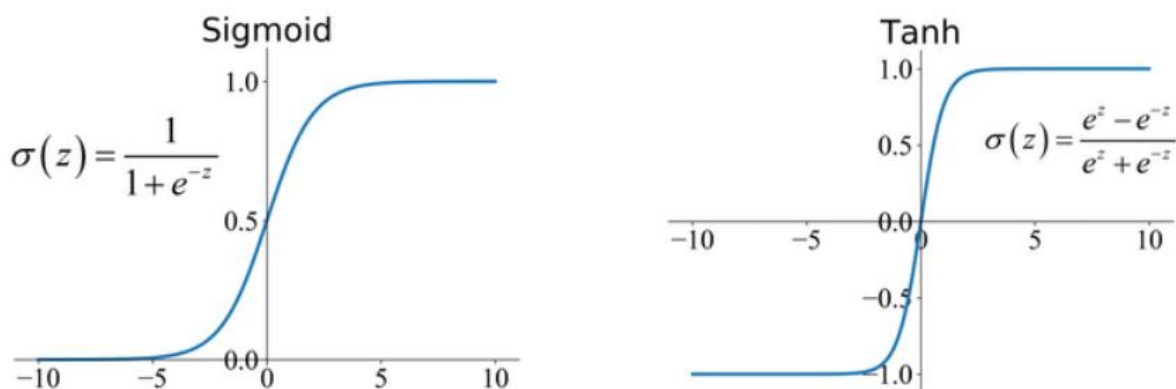


Fig12. Example of activation function

– Encoding vector

In this lab, an encoding vector refers to a mathematical vector used to represent images of information. This vector captures essential features of the input data and is commonly employed in fields such as machine learning.

– L1 distance

L1 distance also known as Manhattan distance. It is defined as the sum of the absolute differences between the corresponding coordinates of the two points.

Formula of L1 distance

$$L1 \text{ distance} = \sum_{i=1}^n |p_i - q_i|$$

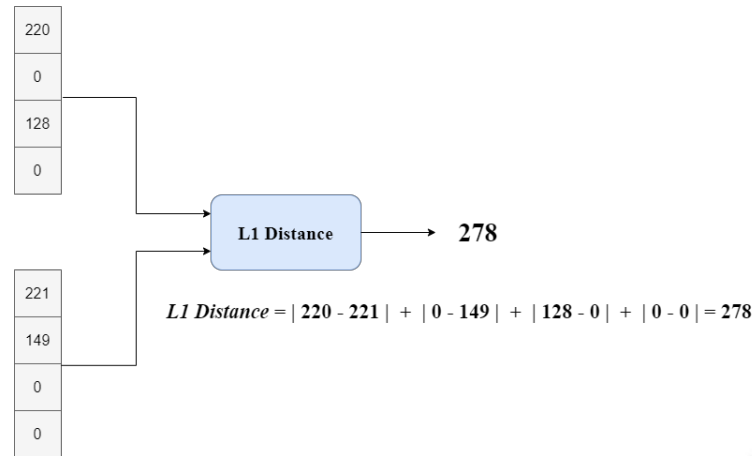


Fig10. Example of L1 distance

Inputs and Outputs

The following are the definitions of input signals

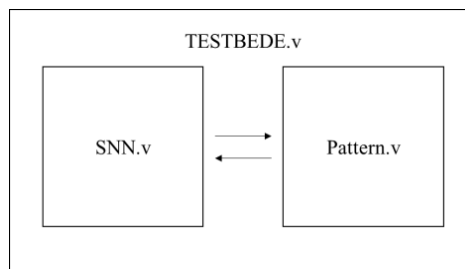
Input Signals	Bit Width	Definition
clk	1	Clock.
rst_n	1	Asynchronous active-low reset.
in_valid	1	High when all input is valid.
Img	32	The image signals which sent in raster ordering. The arithmetic representation follows the IEEE-754 floating number format. (Range: $\mp 0.5 \sim 255.0$)
Kernel	32	The kernel signals which sent in raster ordering. The arithmetic representation follows the IEEE-754 floating number format. (Range: $\mp 0 \sim 0.5$)
Weight	32	The weight signals which sent in raster ordering. The arithmetic representation follows the IEEE-754 floating number format. (Range: \mp

1. The input signal **Img** is delivered in raster scan order for **96 cycles** continuously. When **in_valid** is low, input is tied to unknown state.
2. The input signal **Kernel** is delivered in raster scan order for **27 cycles** continuously.
3. The input signal **Weight** is delivered in raster scan order for **4 cycles** continuously.
4. The input signal **Opt** is delivered for **only 1 cycle during the first cycle of in_valid tied high**. After 1 cycle, input is tied to unknown state.
5. All input signals are synchronized at negative edge of the clock.
6. The output signal **out** must be delivered for **only 1 cycle**, and **out_valid** should be **high** simultaneously.
7. The **out** signal should be **zero** when **out_valid** is low.
8. The **out_valid** cannot overlap with **in_valid** at any time.
9. **Please follow the parameter TA set, or you might fail in this lab.**
10. **You don't need to worry about infinity in the calculation process.**

Specifications

1. Top module name: SNN (File name: SNN.v)
2. **You have to check an error under 0.002 for the result after converting to float number. If the error is higher than the value, you will fail this lab.**
3. **It is asynchronous reset and active-low architecture. If you use synchronous reset (considering reset after clock starting) in your design, you may fail to reset signals.**
4. The reset signal (rst_n) would be given only once at the beginning of simulation. All output signals should be reset after the reset signal is asserted.
5. The **out** should be reset after your **out_valid** is pulled down.
6. The execution latency is limited in **1000 cycles**. The latency is the clock cycles between the falling edge of the **in_valid** and the rising edge of the first **out_valid**.
7. The area is limited in **5500000**. Also, the synthesis time should be less than **3 hours**.
8. You can adjust your clock period by yourself, but the maximum period is **50 ns**. The precision of clock period is 0.1, for example, 4.5 is allowed, 4.55 is not allowed.
9. The input delay is set to **0.5*(clock period)**.
10. The output delay is set to **0.5*(clock period)**, and the output loading is set to **0.05**.
11. The synthesis result of data type **cannot** include any **latches**.
12. After synthesis, you can check SNN.area and SNN.timing. The area report is valid when the slack in the end of timing report should be **non-negative (MET)**.
13. **In this lab, you must use at least one IEEE floating point number IP from Designware. We will check it at SNN.resource in 02_SYN/Report/.**

Block Diagram



Grading Policy

1. Function Validity: 70%
2. Performance: 30 %
 - Area * Computation time: 30%
 - Computation time = Latency * clock cycle time

Note

1. Please submit following files under 09_SUBMIT before 12:00 at noon on October. 16:

- **SNN.v**
- If uploaded files **violate the naming rule**, you will get **5 deduct points**.
- In this lab, you can adjust your clock cycle time.
- The 2nd demo deadline is **12:00 at noon on October.18** .
- Check whether there is any wire / reg / submodule name called “error”, “fail”, “pass”, “congratulation”, “latch”, “DW_fp”, if you used, you will fail the lab.

2. Template folders and reference commands:

01_RTL/	(RTL simulation)	./01_run_vcs_rtl
02_SYN/	(Synthesis)	./01_run_dc_shell
(Check if there is any latch in your design in syn.log)		
(Check the timing of design in /Report/SNN.timing)		
03_GATE /	(Gate-level simulation)	./01_run_vcs_gate
09_SBMIT /	(submit your files)	./00_tar
		./01_submit
		./02_check

※You should make sure the two clock period values identical in 00_TESTBED/Pattern.v && /02_SYN/syn.tcl:

```
`define CYCLE_TIME      50.0
`define SEED_NUMBER     28825252
`define PATTERN_NUMBER  1000
```

```
#=====
# (A) Global Parameters
#=====
set DESIGN "SNN"
set CYCLE 50
```

Sample Waveform

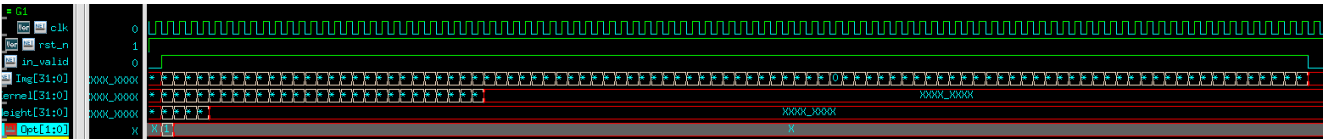


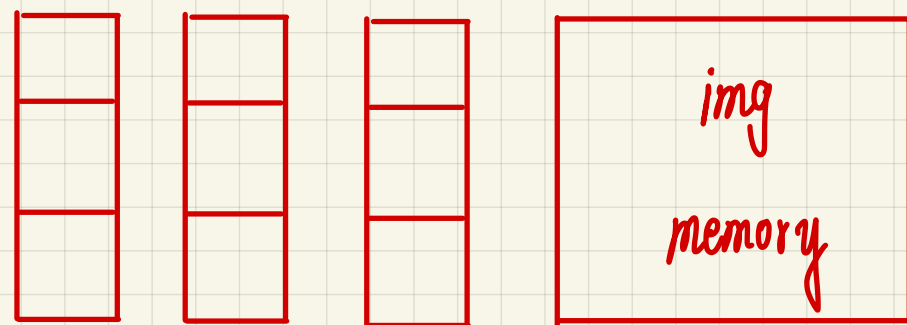
Fig1. Input waveform



Fig2. Output waveform



T



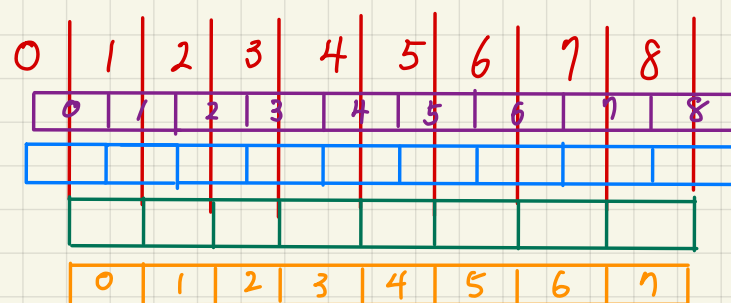
$$\text{col} \% 4 = 0$$

$$\text{row} \% 4 = 0$$

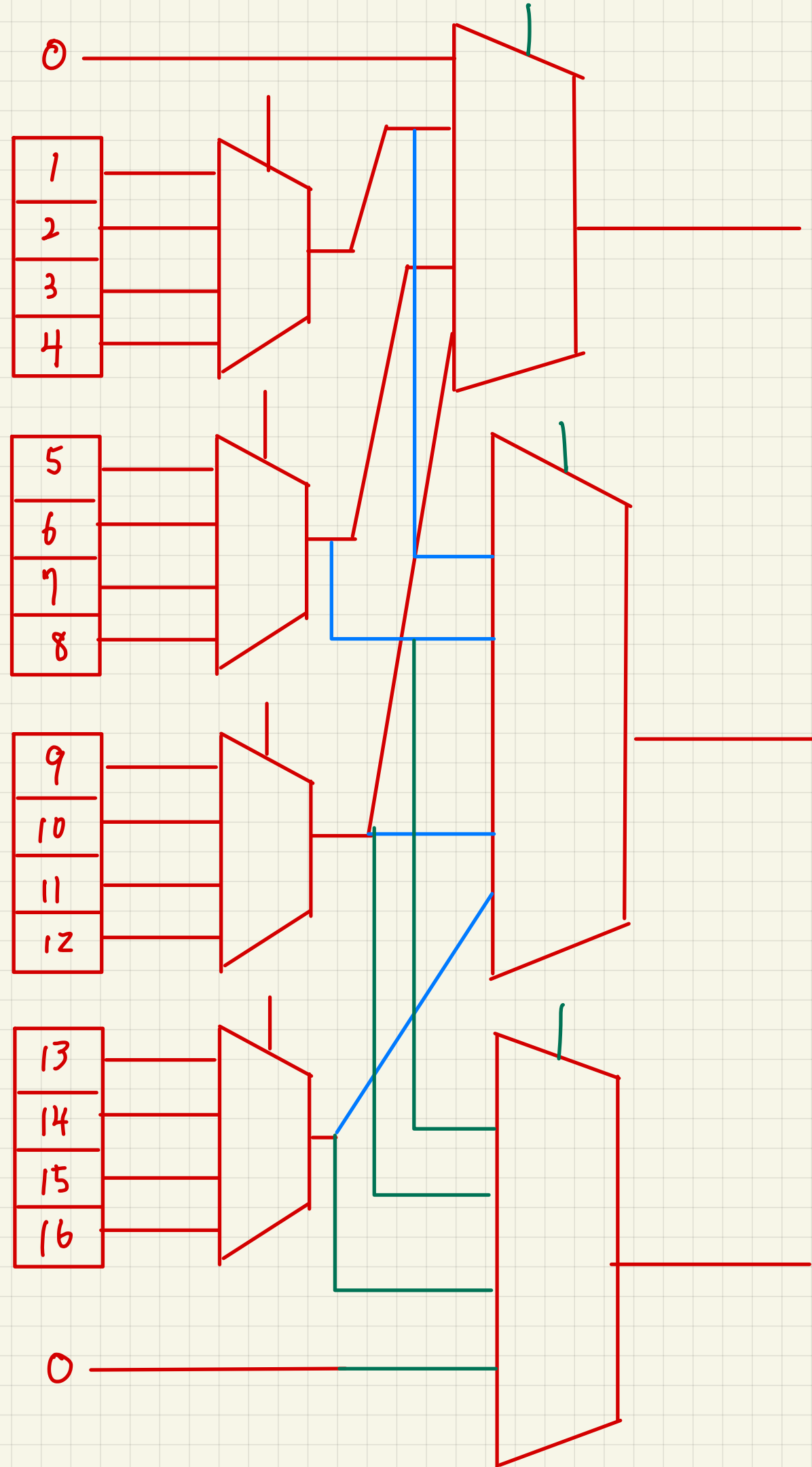
1
2
3

0	0	0	0	0	0
0	1	1	2	1	0
0	3	2	1	1	0
0	1	2	3	1	0
0	4	3	2	2	0
0	0	0	0	0	0

input_count
image
kernel
kernel_save
image_save



$$11 + 3 + 4 + 2$$



1	4	1
3	2	3
1	2	2

$$2 + 3 + 6 + 4 = 15$$
$$= 20$$

90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105

74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89

58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73

42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57

26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41

9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25

9	10	11	12	13	14	15
---	----	----	----	----	----	----

dp3 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15

000	000	000	000	001	012	123	234	345	456	567	678	789	8910	91011	101112	11120
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	-------	--------	-------

001 012 123 234 345 456 567 678 789 8910 91011 101112 111213 121314 131415 141516 151617

1. 005 001 003 007 009 011 013 015 017 019 021 023 025 027 029 031 033 035 037 039 041 043 045 047 049 051 053 055 057 059 061 063 065 067 069 071 073 075 077 079 081 083 085 087 089 091 093 095 097 099 101 103 105 107 109 111 113 115 117 119 121 123 125 127 129 131 133 135 137 139 141 143 145 147 149 151 153 155 157 159 161 163 165 167 169 171 173 175 177 179 181 183 185 187 189 191 193 195 197 199 201 203 205 207 209 211 213 215 217 219 221 223 225 227 229 231 233 235 237 239 241 243 245 247 249 251 253 255 257 259 261 263 265 267 269 271 273 275 277 279 281 283 285 287 289 291 293 295 297 299 301 303 305 307 309 311 313 315 317 319 321 323 325 327 329 331 333 335 337 339 341 343 345 347 349 351 353 355 357 359 361 363 365 367 369 371 373 375 377 379 381 383 385 387 389 391 393 395 397 399 401 403 405 407 409 411 413 415 417 419 421 423 425 427 429 431 433 435 437 439 441 443 445 447 449 451 453 455 457 459 461 463 465 467 469 471 473 475 477 479 481 483 485 487 489 491 493 495 497 499 501 503 505 507 509 511 513 515 517 519 521 523 525 527 529 531 533 535 537 539 541 543 545 547 549 551 553 555 557 559 561 563 565 567 569 571 573 575 577 579 581 583 585 587 589 591 593 595 597 599 601 603 605 607 609 611 613 615 617 619 621 623 625 627 629 631 633 635 637 639 641 643 645 647 649 651 653 655 657 659 661 663 665 667 669 671 673 675 677 679 681 683 685 687 689 691 693 695 697 699 701 703 705 707 709 711 713 715 717 719 721 723 725 727 729 731 733 735 737 739 741 743 745 747 749 751 753 755 757 759 761 763 765 767 769 771 773 775 777 779 781 783 785 787 789 791 793 795 797 799 801 803 805 807 809 811 813 815 817 819 821 823 825 827 829 831 833 835 837 839 841 843 845 847 849 851 853 855 857 859 861 863 865 867 869 871 873 875 877 879 881 883 885 887 889 891 893 895 897 899 901 903 905 907 909 911 913 915 917 919 921 923 925 927 929 931 933 935 937 939 941 943 945 947 949 951 953 955 957 959 961 963 965 967 969 971 973 975 977 979 981 983 985 987 989 991 993 995 997 999 1001 1003 1005 1007 1009 1011 1013 1015 1017 1019 1021 1023 1025 1027 1029 1031 1033 1035 1037 1039 1041 1043 1045 1047 1049 1051 1053 1055 1057 1059 1061 1063 1065 1067 1069 1071 1073 1075 1077 1079 1081 1083 1085 1087 1089 1091 1093 1095 1097 1099 1101 1103 1105 1107 1109 1111 1113 1115 1117 1119 1121 1123 1125 1127 1129 1131 1133 1135 1137 1139 1141 1143 1145 1147 1149 1151 1153 1155 1157 1159 1161 1163 1165 1167 1169 1171 1173 1175 1177 1179 1181 1183 1185 1187 1189 1191 1193 1195 1197 1199 1201 1203 1205 1207 1209 1211 1213 1215 1217 1219 1221 1223 1225 1227 1229 1231 1233 1235 1237 1239 1241 1243 1245 1247 1249 1251 1253 1255 1257 1259 1261 1263 1265 1267 1269 1271 1273 1275 1277 1279 1281 1283 1285 1287 1289 1291 1293 1295 1297 1299 1301 1303 1305 1307 1309 1311 1313 1315 1317 1319 1321 1323 1325 1327 1329 1331 1333 1335 1337 1339 1341 1343 1345 1347 1349 1351 1353 1355 1357 1359 1361 1363 1365 1367 1369 1371 1373 1375 1377 1379 1381 1383 1385 1387 1389 1391 1393 1395 1397 1399 1401 1403 1405 1407 1409 1411 1413 1415 1417 1419 1421 1423 1425 1427 1429 1431 1433 1435 1437 1439 1441 1443 1445 1447 1449 1451 1453 1455 1457 1459 1461 1463 1465 1467 1469 1471 1473 1475 1477 1479 1481 1483 1485 1487 1489 1491 1493 1495 1497 1499 1501 1503 1505 1507 1509 1511 1513 1515 1517 1519 1521 1523 1525 1527 1529 1531 1533 1535 1537 1539 1541 1543 1545 1547 1549 1551 1553 1555 1557 1559 1561 1563 1565 1567 1569 1571 1573 1575 1577 1579 1581 1583 1585 1587 1589 1591 1593 1595 1597 1599 1601 1603 1605 1607 1609 1611 1613 1615 1617 1619 1621 1623 1625 1627 1629 1631 1633 1635 1637 1639 1641 1643 1645 1647 1649 1651 1653 1655 1657 1659 1661 1663 1665 1667 1669 1671 1673 1675 1677 1679 1681 1683 1685 1687 1689 1691 1693 1695 1697 1699 1701 1703 1705 1707 1709 1711 1713 1715 1717 1719 1721 1723 1725 1727 1729 1731 1733 1735 1737 1739 1741 1743 1745 1747 1749 1751 1753 1755 1757 1759 1761 1763 1765 1767 1769 1771 1773 1775 1777 1779 1781 1783 1785 1787 1789 1791 1793 1795 1797 1799 1801 1803 1805 1807 1809 1811 1813 1815 1817 1819 1821 1823 1825 1827 1829 1831 1833 1835 1

dp3 3

Ques 3

sums

col	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1
-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

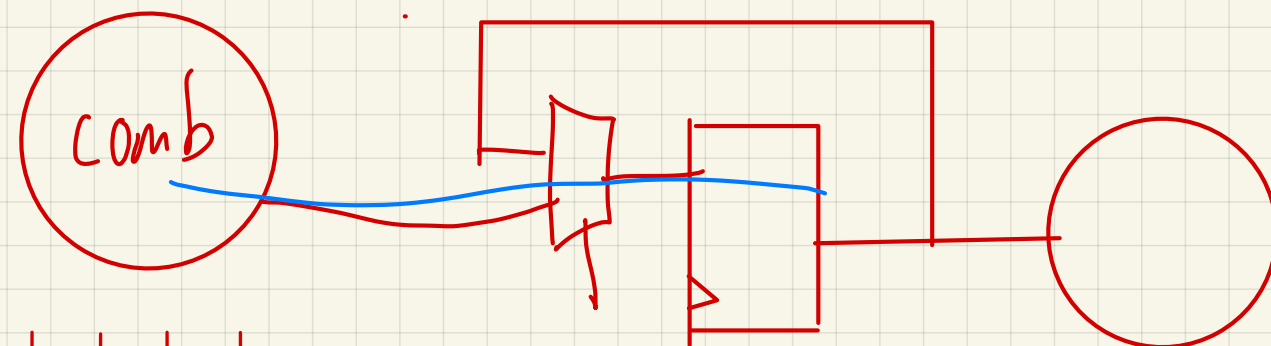
row 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3 0 0

per Freitag

[illegible]

cmp	8				0	1	5	1	5	0		0	10	11	12
-----	--------------	--	--	--	---	---	---	---	---	---	--	---	----	----	----

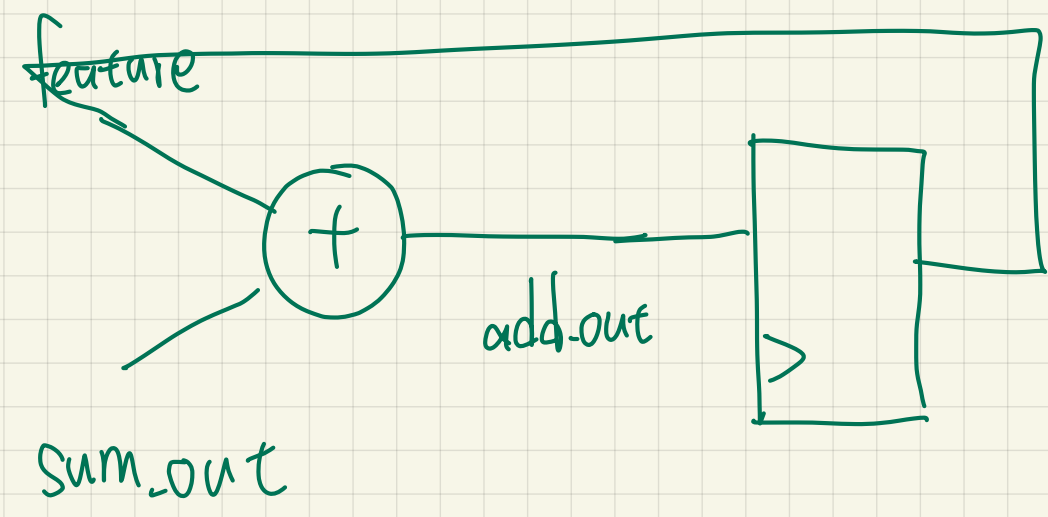
61	62
----	----



[illegible]

$$-1.25 + 6 - 1.5 = -8.75$$

$$2.5 - 13.75 - 16 =$$



wei

$$0.5 \quad -2$$

$$1 \quad 2$$

$$-0.5 \quad -4 \quad -4 \quad -8$$

chl

0	0	0	0	0	0
0	-0.5	1	-4	-1	0
0	4	-4	2	-0.5	0
0	-2	-0.5	-0.5	2	0
0	0.5	-0.5	0.5	2	0
0	0	0	0	0	0

-2	-0.5	-0.5	0.5
-4	-1	2	4
0.5	-2	2	4
-4	2	1	-2

-1	-2	-1	4
4	2	2	2
0.5	1	2	-2
-1	-2	2	-4

$$\begin{matrix} 2 & 2 & 4 \\ 2 & 0.5 & -1 \\ -2 & -2 & -0.5 \end{matrix}$$

$$\begin{matrix} -1 & 1 & -2 \\ 4 & -0.5 & 2 \\ 0.5 & -1 & -2 \end{matrix}$$

$$\begin{matrix} -0.5 & 1 & 2 \\ -4 & 0.5 & 2 \\ -4 & -1 & 0.5 \end{matrix}$$

