Handwritten Alphabet Recognition

Arun Kumar, Raj Vardhan Bundela, Rohit Kumar

Supervisor: Dr. Gaurav Kaushal

ABV-IIITM, Gwalior

November 13, 2022

विश्वजीवनामृतं ज्ञानम्

Table of Contents

1. Introduction
2. Objective
3. Experimental Setup
4. Experimental Results
5. Performance Estimates
6. Memory
7. Utilization Estimates
8. Vivado Results
9. Conclusion

Introduction

- **Handwritten alphabet classification** can be confusing and sometimes an erroneous task as different people write alphabet characters in different **shapes and sizes**.
- Each alphabet's character can is represented as an **image** and hence can be processed for finding **similar type of patterns** and hence can be used for classification.

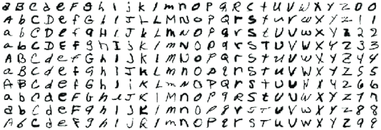


Figure 1: Handwritten characters

Objective

- To train an ANN (Artificial Neural Network) model for prediction of handwritten alphabets.
- Defining the required hardware configuration for the same in the industry.

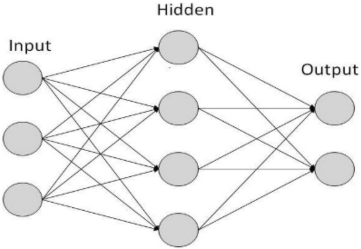


Figure 2: General ANN network

Experimental setup

Dataset:**NIST**

- The National Institute of Standards and Technology produced the dataset **(NIST)**. The example png images in the NIST Special Database 19 total about 0.7 million. The current model has only received training for capital letters (A-Z) which are about **0.4 million images** in total.
- The **3 Densely layers: input, hidden, and output** performing computation and push the output forward with activation function set as 'Sigmoid' for all the layers.
- Output layer contains **26 neurons**, and is wrapped with 'Sigmoid' activation function.

Experimental setup

Dataset:**NIST**

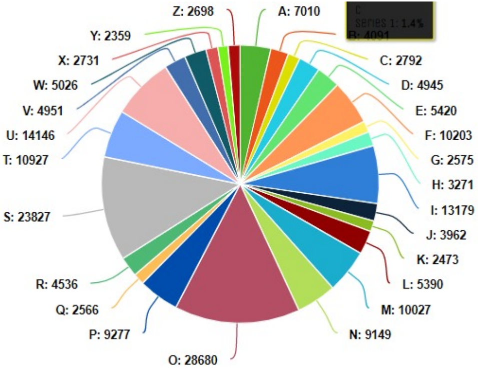


Figure 3: Distribution of Dataset

Model Function

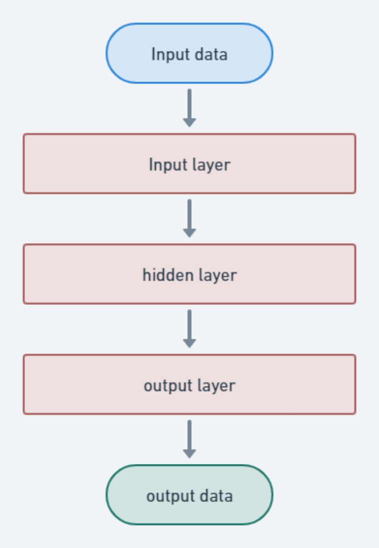


Figure 4: Technical Flowchart

Experimental result

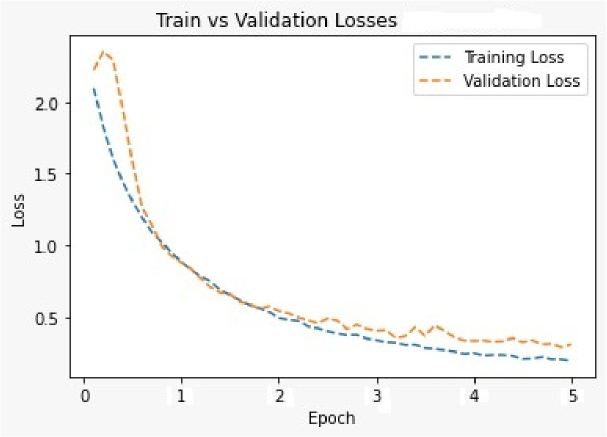


Figure 5: Train vs Validation losses

Experimental result

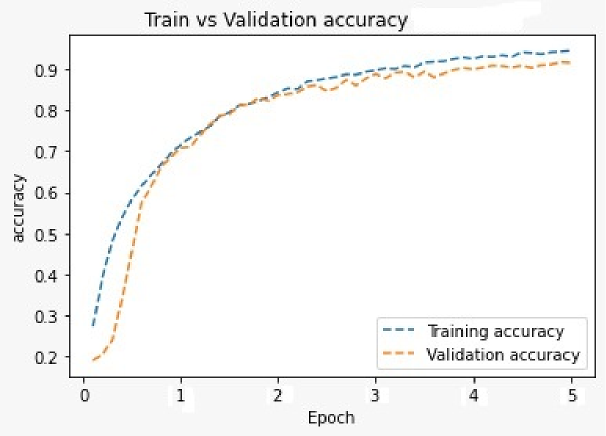
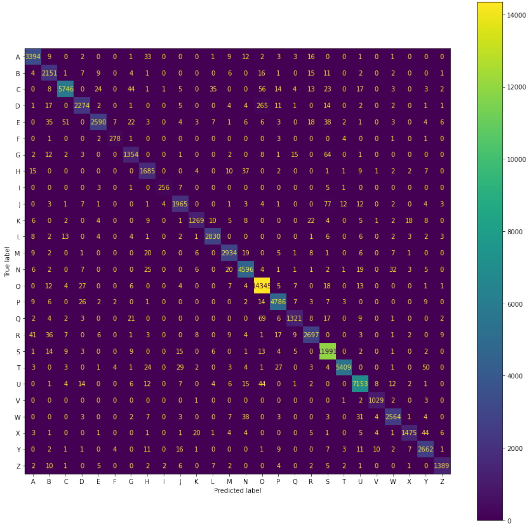


Figure 6: Train vs Validation accuracy

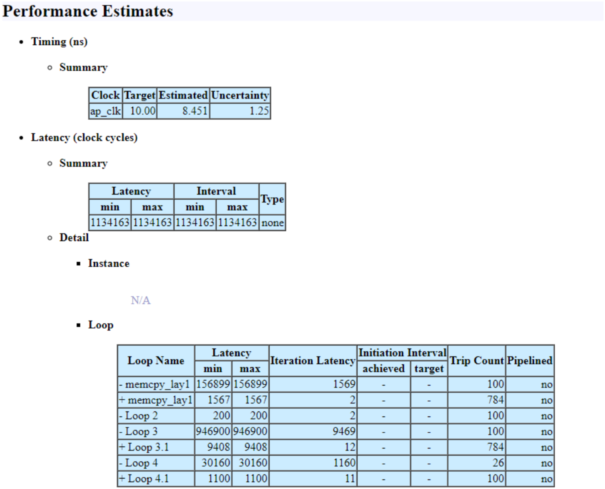
Experimental result



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	3394	9	0	2	0	0	1	33	0	0	0	1	9	12	2	3	3	16	0	0	1	0	1	0	0	0
B	4	2151	1	7	9	0	4	1	0	0	0	0	6	0	16	1	0	15	11	0	2	0	2	0	0	1
C	0	8	5746	0	24	0	44	1	1	5	0	35	0	0	56	14	4	13	23	0	17	0	3	0	3	2
D	1	17	0	2274	2	0	1	0	0	5	0	0	4	4	265	11	1	0	14	0	2	0	2	0	1	1
E	0	35	51	0	2590	7	22	3	0	4	3	7	1	6	6	3	0	18	38	2	1	0	3	0	4	6
F	0	1	0	0	2	278	1	0	0	0	0	0	0	0	0	3	0	0	0	4	0	0	1	0	1	0
G	2	12	2	3	0	0	1354	0	0	1	0	0	2	0	8	1	15	0	64	0	1	0	0	0	0	0
H	15	0	0	0	0	0	0	1685	0	0	4	0	10	37	0	2	0	0	1	1	9	1	2	2	7	0
I	0	0	0	0	3	0	1	0	256	7	0	0	0	0	0	0	0	0	5	1	0	0	0	0	0	0
J	0	3	1	7	1	0	0	1	4	1965	0	0	1	3	4	1	0	0	77	12	12	0	2	0	4	3
K	6	0	2	0	4	0	0	9	0	1	1269	10	5	8	0	0	0	22	4	0	5	1	2	18	8	0
L	8	2	13	0	4	0	4	1	0	2	1	2830	0	0	0	0	0	1	6	0	6	0	2	3	2	3
M	9	2	0	1	0	0	0	20	0	0	6	0	2934	19	0	5	1	8	1	0	6	0	7	1	0	0
N	6	2	0	7	0	0	0	25	0	0	6	0	20	4596	4	1	1	1	2	1	19	0	32	3	5	0
O	0	12	4	27	0	0	6	0	0	4	0	0	7	4	14345	5	7	0	18	0	13	0	0	0	1	1
P	9	6	0	26	2	2	0	1	0	0	0	0	0	2	14	4786	7	3	7	3	0	0	0	0	9	0
Q	2	4	2	3	0	0	21	0	0	0	0	0	0	0	69	6	1321	8	17	0	9	0	1	0	0	2
R	41	36	7	0	6	0	1	3	0	0	8	0	9	4	1	17	9	2697	0	0	3	0	1	2	0	9
S	1	14	9	3	0	0	9	0	0	15	0	6	0	1	13	4	5	0	11991	0	2	0	1	0	2	0
T	3	0	3	0	1	4	1	24	0	29	2	0	3	4	1	27	0	3	4	5409	0	0	1	0	50	0
U	0	1	4	14	0	0	6	12	0	7	0	4	6	15	44	0	1	2	0	0	7153	8	12	2	1	0
V	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	2	1029	2	0	3	0
W	0	0	0	3	0	0	2	7	0	3	0	0	7	38	0	3	0	0	3	0	31	4	2564	1	4	0
X	3	1	0	0	1	0	0	1	0	1	20	1	4	4	0	0	0	5	1	0	5	4	1	1475	44	6
Y	0	2	1	1	0	4	0	11	0	16	1	0	0	0	1	9	0	0	7	3	11	10	2	7	2662	1
Z	2	10	1	0	5	0	0	2	2	6	0	7	2	0	0	4	0	2	5	2	1	0	0	1	0	1389

Figure 7: Confusion Matrix

Performance Estimates



Performance Estimates

- **Timing (ns)**
  - **Summary**

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.451	1.25

- **Latency (clock cycles)**
  - **Summary**

Latency		Interval		Type
min	max	min	max	
1134163	1134163	1134163	1134163	none

  - **Detail**
    - **Instance**

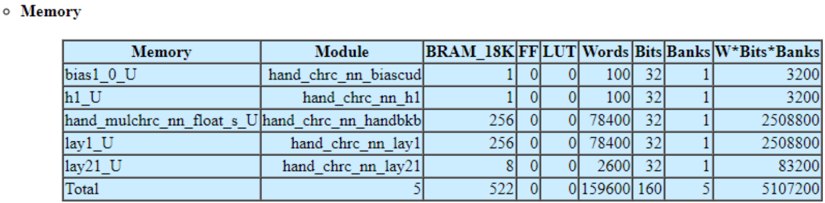
N/A

    - **Loop**

Loop Name	Latency		Iteration Latency	Initiation Interval		Trip Count	Pipelined
	min	max		achieved	target		
- memcpy_lay1	156899	156899	1569	-	-	100	no
+ memcpy_lay1	1567	1567	2	-	-	784	no
- Loop 2	200	200	2	-	-	100	no
- Loop 3	946900	946900	9469	-	-	100	no
+ Loop 3.1	9408	9408	12	-	-	784	no
- Loop 4	30160	30160	1160	-	-	26	no
+ Loop 4.1	1100	1100	11	-	-	100	no

Figure 8: Performance Estimates

Memory

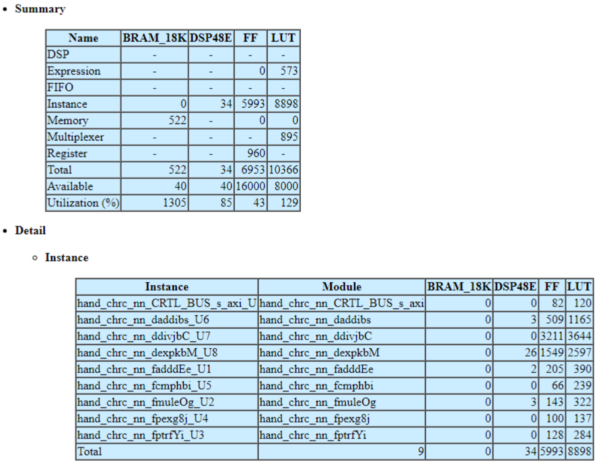


- **Memory**

Memory	Module	BRAM_18K	FF	LUT	Words	Bits	Banks	W*Bits*Banks
bias1_0_U	hand_chrc_nn_biascud	1	0	0	100	32	1	3200
h1_U	hand_chrc_nn_h1	1	0	0	100	32	1	3200
hand_mulchrc_nn_float_s_U	hand_chrc_nn_handbkb	256	0	0	78400	32	1	2508800
lay1_U	hand_chrc_nn_lay1	256	0	0	78400	32	1	2508800
lay21_U	hand_chrc_nn_lay21	8	0	0	2600	32	1	83200
Total	5	522	0	0	159600	160	5	5107200

Figure 9: Memory

Utilization Estimates



- **Summary**

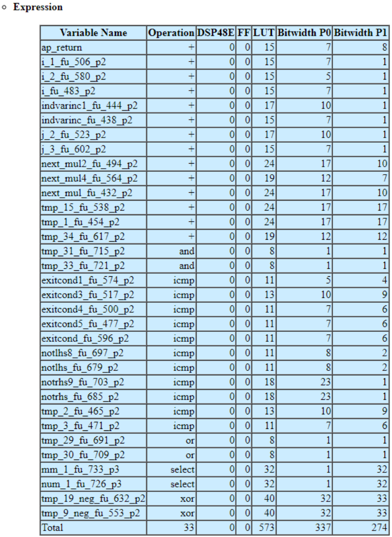
Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	573
FIFO	-	-	-	-
Instance	0	34	5993	8898
Memory	522	-	0	0
Multiplexer	-	-	-	895
Register	-	-	960	-
Total	522	34	6953	10366
Available	40	40	16000	8000
Utilization (%)	1305	85	43	129

- **Detail**
  - **Instance**

Instance	Module	BRAM_18K	DSP48E	FF	LUT
hand_chrc_nn_CRTL_BUS_s_axi_U	hand_chrc_nn_CRTL_BUS_s_axi	0	0	82	120
hand_chrc_nn_daddibs_U6	hand_chrc_nn_daddibs	0	3	509	1165
hand_chrc_nn_ddivjbC_U7	hand_chrc_nn_ddivjbC	0	0	3211	3644
hand_chrc_nn_dexpkbM_U8	hand_chrc_nn_dexpkbM	0	26	1549	2597
hand_chrc_nn_fadddEe_U1	hand_chrc_nn_fadddEe	0	2	205	390
hand_chrc_nn_fcmphbi_U5	hand_chrc_nn_fcmphbi	0	0	66	239
hand_chrc_nn_fmuleOg_U2	hand_chrc_nn_fmuleOg	0	3	143	322
hand_chrc_nn_fpexg8j_U4	hand_chrc_nn_fpexg8j	0	0	100	137
hand_chrc_nn_fptrfYi_U3	hand_chrc_nn_fptrfYi	0	0	128	284
Total	9	0	34	5993	8898

Figure 10: Utilization Estimates

Vivado Results

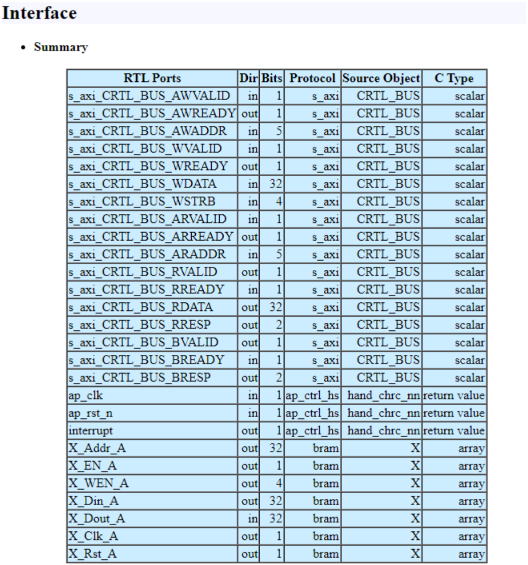


- Expression

Variable Name	Operation	DSP48E	FF	LUT	Bitwidth P0	Bitwidth P1
ap_return	+	0	0	15	7	8
i_1_fu_506_p2	+	0	0	15	7	1
i_2_fu_580_p2	+	0	0	15	5	1
i_fu_483_p2	+	0	0	15	7	1
indvarinc1_fu_444_p2	+	0	0	17	10	1
indvarinc_fu_438_p2	+	0	0	15	7	1
j_2_fu_523_p2	+	0	0	17	10	1
j_3_fu_602_p2	+	0	0	15	7	1
next_mul2_fu_494_p2	+	0	0	24	17	10
next_mul4_fu_564_p2	+	0	0	19	12	7
next_mul_fu_432_p2	+	0	0	24	17	10
tmp_15_fu_538_p2	+	0	0	24	17	17
tmp_1_fu_454_p2	+	0	0	24	17	17
tmp_34_fu_617_p2	+	0	0	19	12	12
tmp_31_fu_715_p2	and	0	0	8	1	1
tmp_33_fu_721_p2	and	0	0	8	1	1
exitcond1_fu_574_p2	icmp	0	0	11	5	4
exitcond3_fu_517_p2	icmp	0	0	13	10	9
exitcond4_fu_500_p2	icmp	0	0	11	7	6
exitcond5_fu_477_p2	icmp	0	0	11	7	6
exitcond_fu_596_p2	icmp	0	0	11	7	6
notlhs8_fu_697_p2	icmp	0	0	11	8	2
notlhs_fu_679_p2	icmp	0	0	11	8	2
notrhs9_fu_703_p2	icmp	0	0	18	23	1
notrhs_fu_685_p2	icmp	0	0	18	23	1
tmp_2_fu_465_p2	icmp	0	0	13	10	9
tmp_3_fu_471_p2	icmp	0	0	11	7	6
tmp_29_fu_691_p2	or	0	0	8	1	1
tmp_30_fu_709_p2	or	0	0	8	1	1
mm_1_fu_733_p3	select	0	0	32	1	32
num_1_fu_726_p3	select	0	0	32	1	32
tmp_19_neg_fu_632_p2	xor	0	0	40	32	33
tmp_9_neg_fu_553_p2	xor	0	0	40	32	33
Total	33	0	0	573	337	274

Figure 11: Expressions

Vivado Results

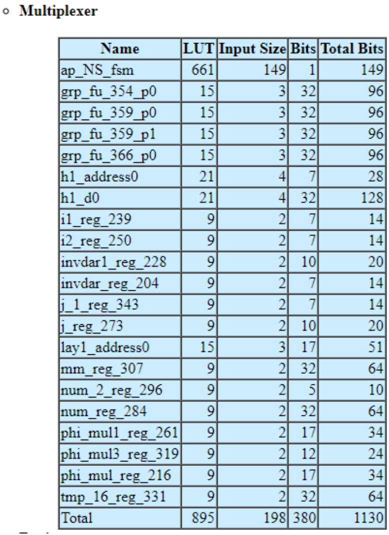


Interface

- **Summary**

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
s_axi_CRTL_BUS_AWVALID	in	1	s_axi	CRTL_BUS	scalar
s_axi_CRTL_BUS_AWREADY	out	1	s_axi	CRTL_BUS	scalar
s_axi_CRTL_BUS_AWADDR	in	5	s_axi	CRTL_BUS	scalar
s_axi_CRTL_BUS_WVALID	in	1	s_axi	CRTL_BUS	scalar
s_axi_CRTL_BUS_WREADY	out	1	s_axi	CRTL_BUS	scalar
s_axi_CRTL_BUS_WDATA	in	32	s_axi	CRTL_BUS	scalar
s_axi_CRTL_BUS_WSTRB	in	4	s_axi	CRTL_BUS	scalar
s_axi_CRTL_BUS_ARVALID	in	1	s_axi	CRTL_BUS	scalar
s_axi_CRTL_BUS_ARREADY	out	1	s_axi	CRTL_BUS	scalar
s_axi_CRTL_BUS_ARADDR	in	5	s_axi	CRTL_BUS	scalar
s_axi_CRTL_BUS_RVALID	out	1	s_axi	CRTL_BUS	scalar
s_axi_CRTL_BUS_RREADY	in	1	s_axi	CRTL_BUS	scalar
s_axi_CRTL_BUS_RDATA	out	32	s_axi	CRTL_BUS	scalar
s_axi_CRTL_BUS_RRESP	out	2	s_axi	CRTL_BUS	scalar
s_axi_CRTL_BUS_BVALID	out	1	s_axi	CRTL_BUS	scalar
s_axi_CRTL_BUS_BREADY	in	1	s_axi	CRTL_BUS	scalar
s_axi_CRTL_BUS_BRESP	out	2	s_axi	CRTL_BUS	scalar
ap_clk	in	1	ap_ctrl_hs	hand_chrc_nn	return value
ap_rst_n	in	1	ap_ctrl_hs	hand_chrc_nn	return value
interrupt	out	1	ap_ctrl_hs	hand_chrc_nn	return value
X_Addr_A	out	32	bram	X	array
X_EN_A	out	1	bram	X	array
X_WEN_A	out	4	bram	X	array
X_Din_A	out	32	bram	X	array
X_Dout_A	in	32	bram	X	array
X_Clk_A	out	1	bram	X	array
X_Rst_A	out	1	bram	X	array

Vivado Results

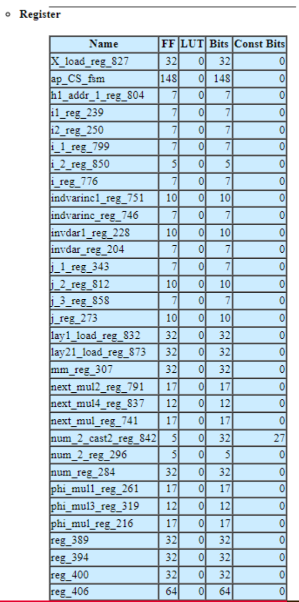


- **Multiplexer**

Name	LUT	Input Size	Bits	Total Bits
ap_NS_fsm	661	149	1	149
grp_fu_354_p0	15	3	32	96
grp_fu_359_p0	15	3	32	96
grp_fu_359_p1	15	3	32	96
grp_fu_366_p0	15	3	32	96
h1_address0	21	4	7	28
h1_d0	21	4	32	128
i1_reg_239	9	2	7	14
i2_reg_250	9	2	7	14
invdar1_reg_228	9	2	10	20
invdar_reg_204	9	2	7	14
j_1_reg_343	9	2	7	14
j_reg_273	9	2	10	20
lay1_address0	15	3	17	51
mm_reg_307	9	2	32	64
num_2_reg_296	9	2	5	10
num_reg_284	9	2	32	64
phi_mul1_reg_261	9	2	17	34
phi_mul3_reg_319	9	2	12	24
phi_mul_reg_216	9	2	17	34
tmp_16_reg_331	9	2	32	64
Total	895	198	380	1130

Figure 13: Multiplexers

Vivado Results



- **Register**

Name	FF	LUT	Bits	Const Bits
X_load_reg_827	32	0	32	0
ap_CS_fsm	148	0	148	0
h1_addr_1_reg_804	7	0	7	0
i1_reg_239	7	0	7	0
i2_reg_250	7	0	7	0
i_1_reg_799	7	0	7	0
i_2_reg_850	5	0	5	0
i_reg_776	7	0	7	0
indvarinc1_reg_751	10	0	10	0
indvarinc_reg_746	7	0	7	0
invdar1_reg_228	10	0	10	0
invdar_reg_204	7	0	7	0
j_1_reg_343	7	0	7	0
j_2_reg_812	10	0	10	0
j_3_reg_858	7	0	7	0
j_reg_273	10	0	10	0
lay1_load_reg_832	32	0	32	0
lay21_load_reg_873	32	0	32	0
mm_reg_307	32	0	32	0
next_mul2_reg_791	17	0	17	0
next_mul4_reg_837	12	0	12	0
next_mul_reg_741	17	0	17	0
num_2_cast2_reg_842	5	0	32	27
num_2_reg_296	5	0	5	0
num_reg_284	32	0	32	0
phi_mul1_reg_261	17	0	17	0
phi_mul3_reg_319	12	0	12	0
phi_mul_reg_216	17	0	17	0
reg_389	32	0	32	0
reg_394	32	0	32	0
reg_400	32	0	32	0
reg_406	64	0	64	0

Experimental conclusion

Following are the conclusions drawn from the experiments:

1) The model performs quite well with an accuracy about 97

2) We have also defined for the necessary hardware configurations, its utilisation and the interface it provides.

Thank You