

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут»

Кафедра КЕОА

Лабораторна робота №1
з курсу: «Апаратні прискорювачі обчислень на мікросхемах
програмованої логіки»

Виконав:
студент III-го курсу
ФЕЛ
група ДК-02
Швець О.В
25.10.2022

Київ-2022

Хід роботи

1. В Simulink реалізувати підсистему, що розраховує функцію:

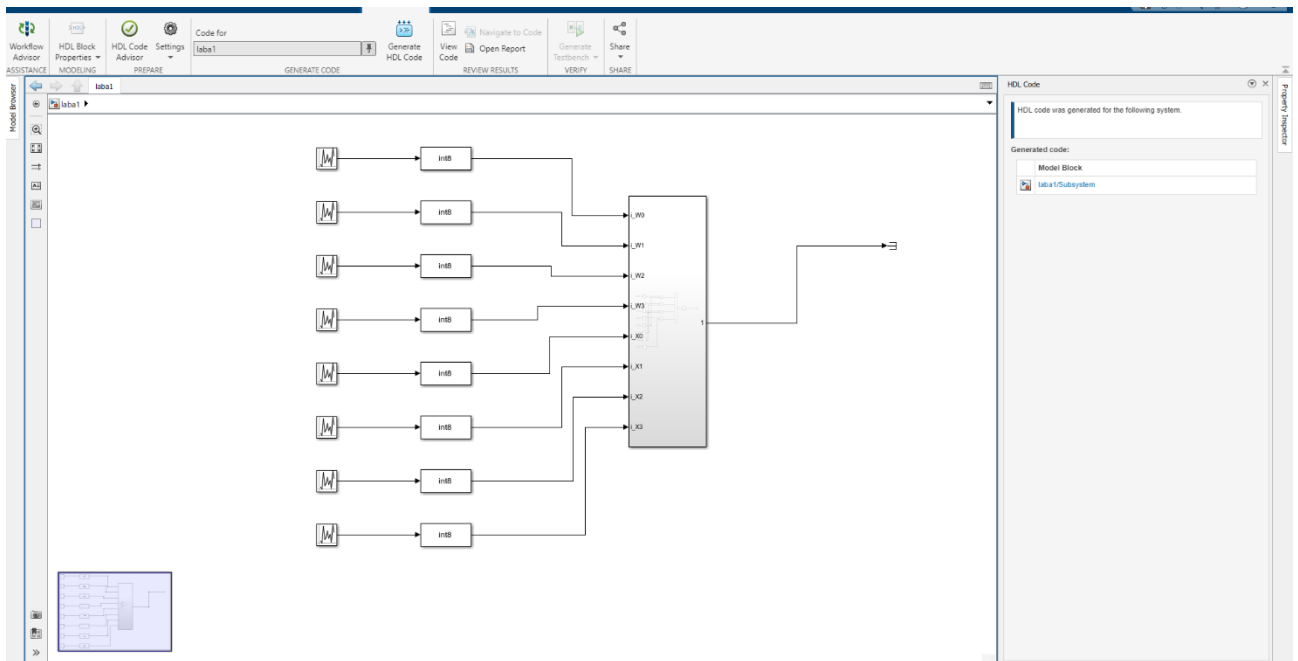
$$Y = W0*X0 + W1*X1 + W2*X2 + W*X3$$

Типи даних входів: int8

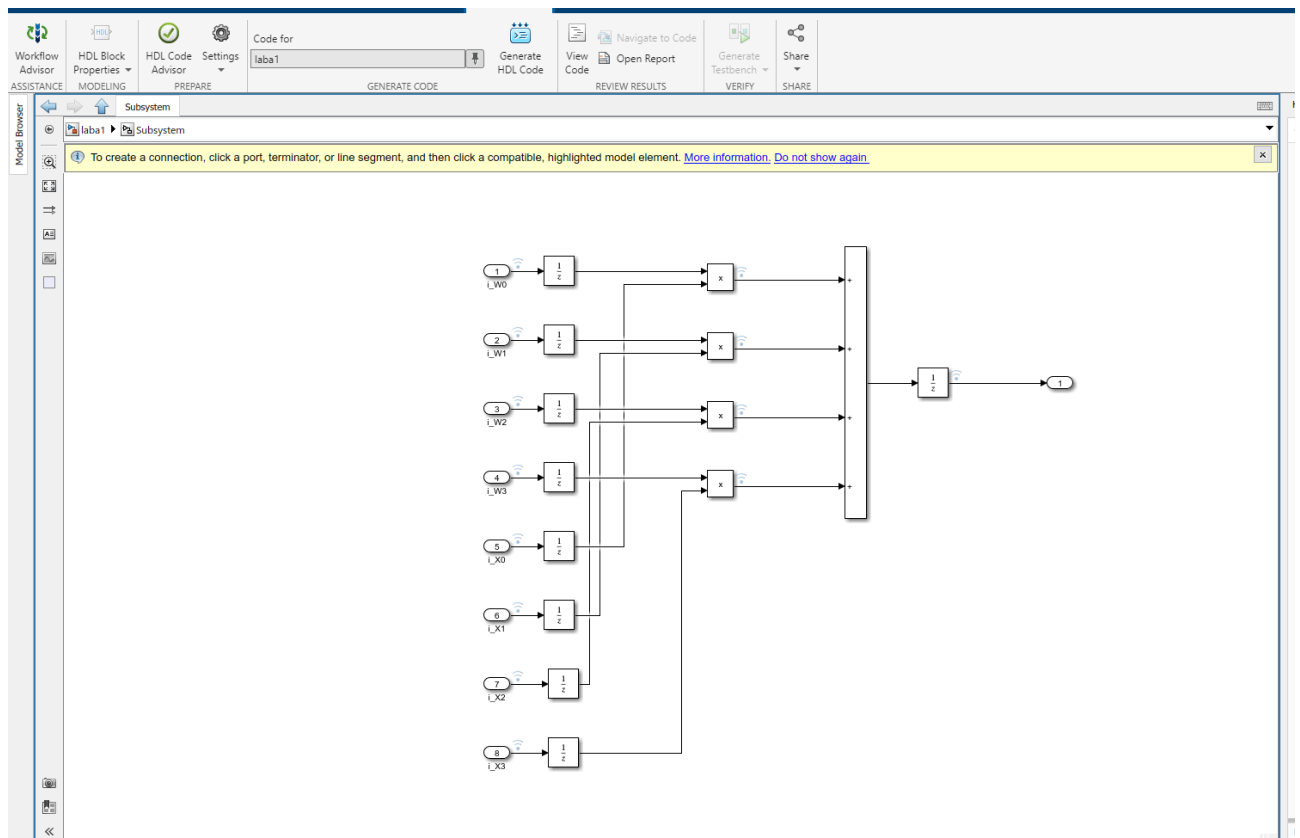
Тип даних виходу: int16

На входах і виході поставити регістри (блок затримки на 1 такт)

Схема має наступний вигляд:



Вигляд всередині блоку Subsystem:



Налаштування першого Uniform Random number:

Block Parameters: Uniform Random Number [X]

Uniform Random Number

Output a uniformly distributed random signal. Output is repeatable for a given seed.

Parameters

Minimum:

0

Maximum:

2^8-1

Seed:

14

Sample time:

1

☒ Interpret vector parameters as 1-D

[?] [OK] [Cancel] [Help] [Apply]

Як вказано у завданні ми збільшуємо значення Seed на 1 при наступних Uniform Random number

Налаштування останнього Uniform Random number:

Block Parameters: Uniform Random Number7

Uniform Random Number

Output a uniformly distributed random signal. Output is repeatable for a given seed.

Parameters

Minimum:

0

Maximum:

2^8-1

Seed:

21

Sample time:

1

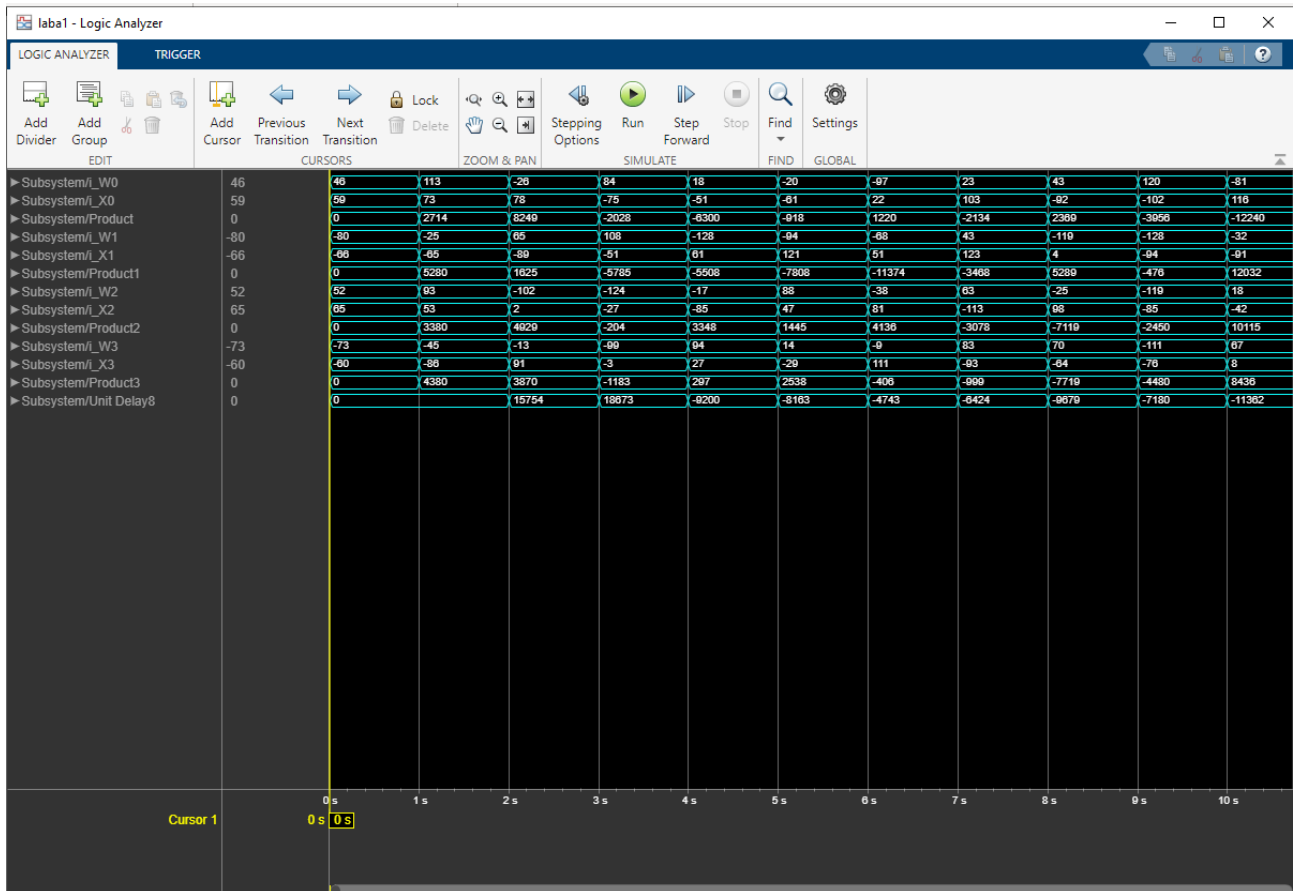
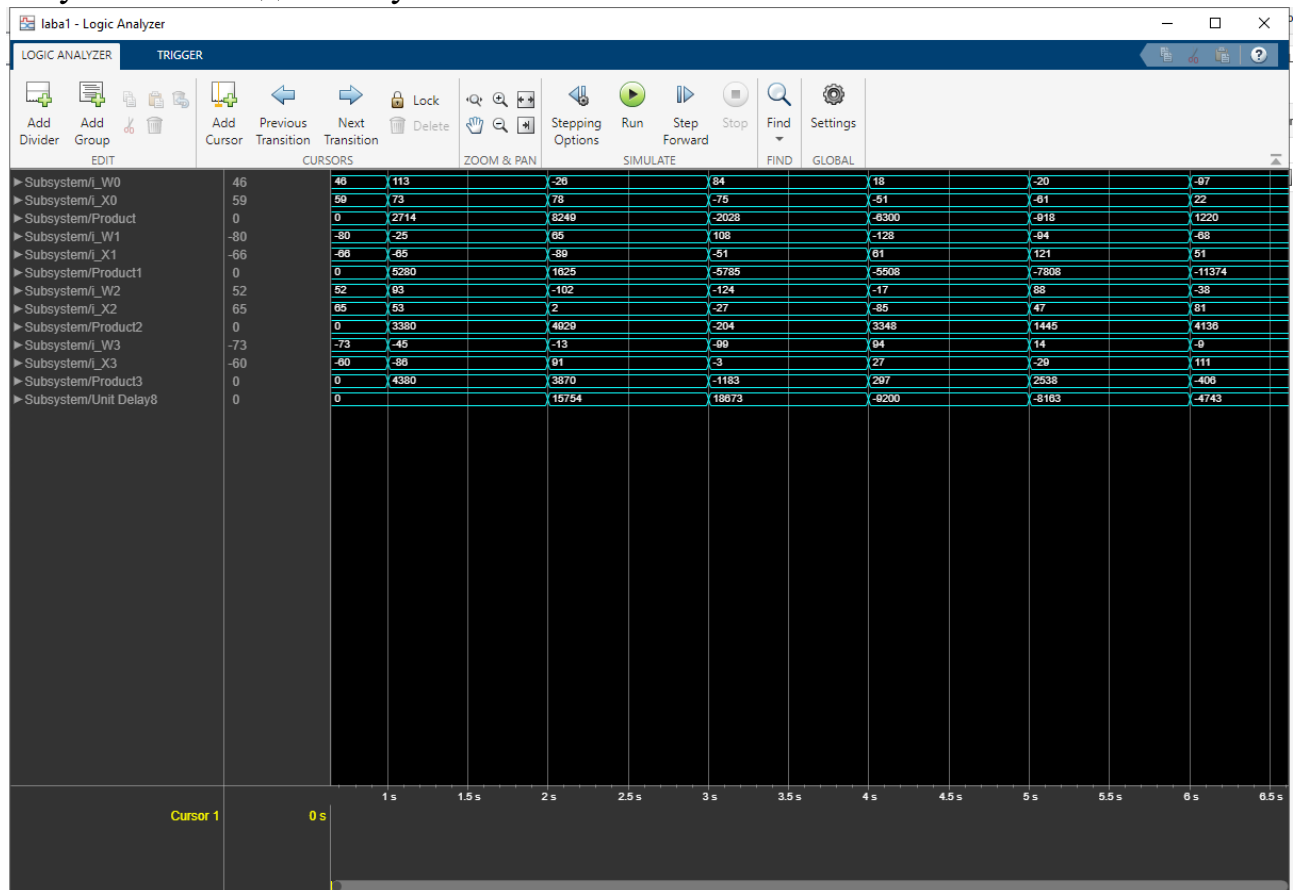
☒ Interpret vector parameters as 1-D

OK Cancel Help Apply

Як можна побачити параметр seed починається з 14 і далі збільшується на 1.

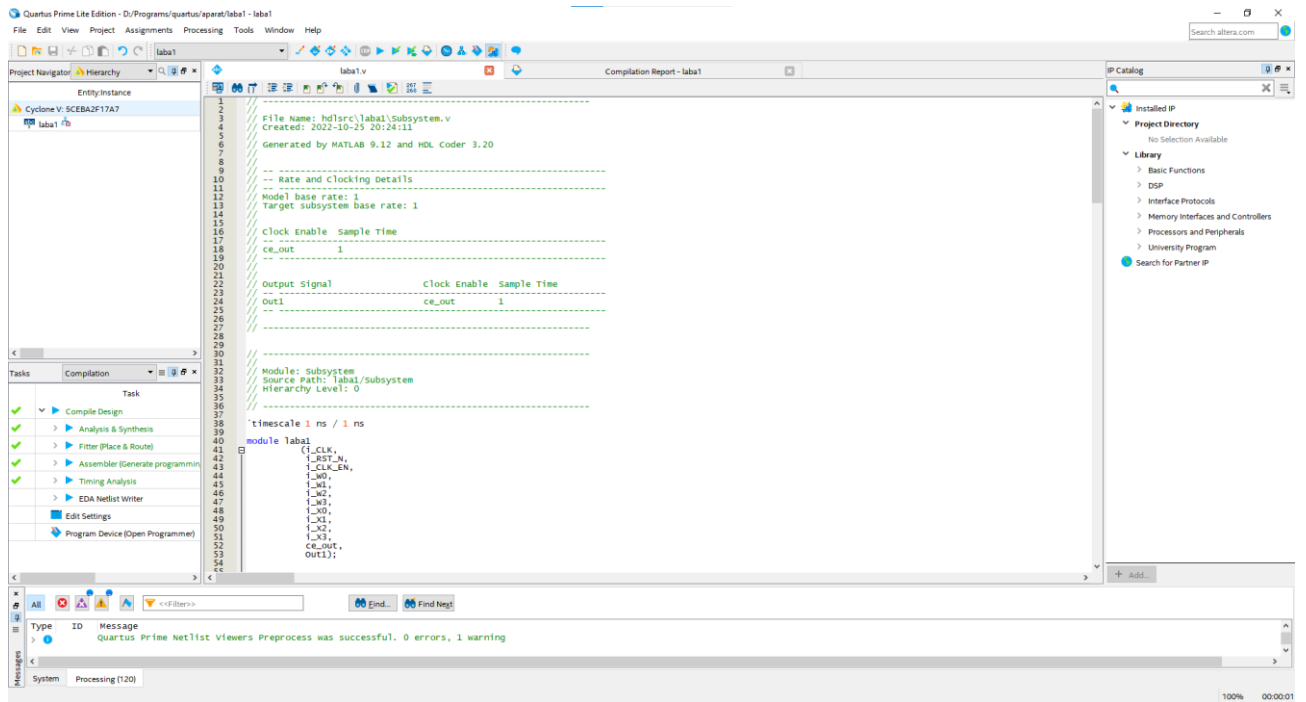
2. В логічному аналізаторі переглянути дані на входах і на виході створеної підсистеми у знаковому десятковому поданні (форматі).

Результат виглядає наступним чином:



$$46*59 + (-80)*(-66) + 52*65 + (-73)*(-60) = 15\,754$$

Згенерований Verilog код:



Project Assignments Processing Tools Window Help

lab1.v

Compilation Report - lab1

Hierarchy Entity:Instance :BA2F17A7

```

49  i_x1,
50  i_x2,
51  i_x3,
52  ce_out,
53  out1);
54
55
56  input  i_CLK;
57  input  i_RST_N;
58  input  i_CLK_EN;
59  input  signed [7:0] i_w0; // int8
60  input  signed [7:0] i_w1; // int8
61  input  signed [7:0] i_w2; // int8
62  input  signed [7:0] i_w3; // int8
63  input  signed [7:0] i_x0; // int8
64  input  signed [7:0] i_x1; // int8
65  input  signed [7:0] i_x2; // int8
66  input  signed [7:0] i_x3; // int8
67  output ce_out;
68  output signed [15:0] out1; // int16
69
70
71  wire enb;
72  reg signed [7:0] Unit_Delay_out1; // int8
73  reg signed [7:0] Unit_Delay2_out1; // int8
74  reg signed [7:0] Unit_Delay3_out1; // int8
75  reg signed [7:0] Unit_Delay4_out1; // int8
76  reg signed [7:0] Unit_Delay5_out1; // int8
77  wire signed [15:0] Product_out1; // int16
78  reg signed [7:0] Unit_Delay6_out1; // int8
79  wire signed [15:0] Product1_out1; // int16
80  wire signed [15:0] Add_stage2_add_temp; // sf16
81  wire signed [16:0] Add_op_stage1; // sf17
82  reg signed [7:0] Unit_Delay7_out1; // int8
83  wire signed [15:0] Product2_out1; // int16
84  wire signed [15:0] Add_stage3_add_cast; // sf16
85  wire signed [15:0] Add_stage3_add_temp; // sf16
86  wire signed [7:0] Add_op_stage2; // sf18
87  reg signed [7:0] Unit_Delay8_out1; // int8
88  wire signed [15:0] Product3_out1; // int16
89  wire signed [15:0] Add_stage4_add_cast; // sf16
90  wire signed [15:0] Add_out1; // int16
91  reg signed [15:0] Unit_Delay8_out1; // int16
92
93
94  assign enb = i_CLK_EN;
95
96  always @(posedge i_CLK or negedge i_RST_N)
97  begin : Unit_Delay_process
98  if (i_RST_N == 1'b0) begin
99  Unit_Delay_out1 <= 8'sb00000000;
100  end
101  else begin
102  if (enb) begin
103  Unit_Delay_out1 <= i_w0;
104  end
105  end
106  end
107
108
109
110  always @(posedge i_CLK or negedge i_RST_N)
111  begin : Unit_Delay1_process
112  if (i_RST_N == 1'b0) begin
113  Unit_Delay1_out1 <= 8'sb00000000;
114  end
115  else begin
116  if (enb) begin
117  Unit_Delay1_out1 <= i_w1;
118  end
119  end
120  end
121
122
123
124  always @(posedge i_CLK or negedge i_RST_N)
125  begin : Unit_Delay2_process
126  if (i_RST_N == 1'b0) begin
127  Unit_Delay2_out1 <= 8'sb00000000;
128  end
129  else begin
130  if (enb) begin
131  Unit_Delay2_out1 <= i_w2;
132  end
133  end
134  end
135
136
137
138  always @(posedge i_CLK or negedge i_RST_N)
139  begin : Unit_Delay3_process
140  if (i_RST_N == 1'b0) begin
141  Unit_Delay3_out1 <= 8'sb00000000;
142  end
143  else begin
144  if (enb) begin
145  Unit_Delay3_out1 <= i_w3;
146  end
147  end
148  end
149
150
151
152  always @(posedge i_CLK or negedge i_RST_N)
153  begin : Unit_Delay4_process
154  if (i_RST_N == 1'b0) begin
155  Unit_Delay4_out1 <= 8'sb00000000;
156  end
157  else begin
158  if (enb) begin
159  Unit_Delay4_out1 <= i_x0;
160  end
161  end
162  end
163
164
165
166  always @(posedge i_CLK or negedge i_RST_N)
167  begin : Unit_Delay5_process
168  if (i_RST_N == 1'b0) begin
169  Unit_Delay5_out1 <= 8'sb00000000;
170  end
171  else begin
172  if (enb) begin
173  Unit_Delay5_out1 <= i_x1;
174  end
175  end
176  end
177
178
179
180  always @(posedge i_CLK or negedge i_RST_N)
181  begin : Unit_Delay6_process
182  if (i_RST_N == 1'b0) begin
183  Unit_Delay6_out1 <= 8'sb00000000;
184  end
185  else begin
186  if (enb) begin
187  Unit_Delay6_out1 <= i_x2;
188  end
189  end
190  end
191
192
193
194  always @(posedge i_CLK or negedge i_RST_N)
195  begin : Unit_Delay7_process
196  if (i_RST_N == 1'b0) begin
197  Unit_Delay7_out1 <= 8'sb00000000;
198  end
199  else begin
200  if (enb) begin
201  Unit_Delay7_out1 <= i_x3;
202  end
203  end
204  end
205
206
207
208  always @(posedge i_CLK or negedge i_RST_N)
209  begin : Unit_Delay8_process
210  if (i_RST_N == 1'b0) begin
211  Unit_Delay8_out1 <= 8'sb00000000;
212  end
213  else begin
214  if (enb) begin
215  Unit_Delay8_out1 <= i_x0;
216  end
217  end
218  end
219
220
221
222  always @(posedge i_CLK or negedge i_RST_N)
223  begin : Unit_Delay9_process
224  if (i_RST_N == 1'b0) begin
225  Unit_Delay9_out1 <= 8'sb00000000;
226  end
227  else begin
228  if (enb) begin
229  Unit_Delay9_out1 <= i_x1;
230  end
231  end
232  end
233
234
235
236  always @(posedge i_CLK or negedge i_RST_N)
237  begin : Unit_Delay10_process
238  if (i_RST_N == 1'b0) begin
239  Unit_Delay10_out1 <= 8'sb00000000;
240  end
241  else begin
242  if (enb) begin
243  Unit_Delay10_out1 <= i_x2;
244  end
245  end
246  end
247
248
249
250  always @(posedge i_CLK or negedge i_RST_N)
251  begin : Unit_Delay11_process
252  if (i_RST_N == 1'b0) begin
253  Unit_Delay11_out1 <= 8'sb00000000;
254  end
255  else begin
256  if (enb) begin
257  Unit_Delay11_out1 <= i_x3;
258  end
259  end
260  end
261
262
263
264  always @(posedge i_CLK or negedge i_RST_N)
265  begin : Unit_Delay12_process
266  if (i_RST_N == 1'b0) begin
267  Unit_Delay12_out1 <= 8'sb00000000;
268  end
269  else begin
270  if (enb) begin
271  Unit_Delay12_out1 <= i_x0;
272  end
273  end
274  end
275
276
277
278  always @(posedge i_CLK or negedge i_RST_N)
279  begin : Unit_Delay13_process
280  if (i_RST_N == 1'b0) begin
281  Unit_Delay13_out1 <= 8'sb00000000;
282  end
283  else begin
284  if (enb) begin
285  Unit_Delay13_out1 <= i_x1;
286  end
287  end
288  end
289
290
291
292  always @(posedge i_CLK or negedge i_RST_N)
293  begin : Unit_Delay14_process
294  if (i_RST_N == 1'b0) begin
295  Unit_Delay14_out1 <= 8'sb00000000;
296  end
297  else begin
298  if (enb) begin
299  Unit_Delay14_out1 <= i_x2;
300  end
301  end
302  end
303
304
305
306  always @(posedge i_CLK or negedge i_RST_N)
307  begin : Unit_Delay15_process
308  if (i_RST_N == 1'b0) begin
309  Unit_Delay15_out1 <= 8'sb00000000;
310  end
311  else begin
312  if (enb) begin
313  Unit_Delay15_out1 <= i_x3;
314  end
315  end
316  end
317
318
319
320  always @(posedge i_CLK or negedge i_RST_N)
321  begin : Unit_Delay16_process
322  if (i_RST_N == 1'b0) begin
323  Unit_Delay16_out1 <= 8'sb00000000;
324  end
325  else begin
326  if (enb) begin
327  Unit_Delay16_out1 <= i_x0;
328  end
329  end
330  end
331
332
333
334  always @(posedge i_CLK or negedge i_RST_N)
335  begin : Unit_Delay17_process
336  if (i_RST_N == 1'b0) begin
337  Unit_Delay17_out1 <= 8'sb00000000;
338  end
339  else begin
340  if (enb) begin
341  Unit_Delay17_out1 <= i_x1;
342  end
343  end
344  end
345
346
347
348  always @(posedge i_CLK or negedge i_RST_N)
349  begin : Unit_Delay18_process
350  if (i_RST_N == 1'b0) begin
351  Unit_Delay18_out1 <= 8'sb00000000;
352  end
353  else begin
354  if (enb) begin
355  Unit_Delay18_out1 <= i_x2;
356  end
357  end
358  end
359
360
361
362  always @(posedge i_CLK or negedge i_RST_N)
363  begin : Unit_Delay19_process
364  if (i_RST_N == 1'b0) begin
365  Unit_Delay19_out1 <= 8'sb00000000;
366  end
367  else begin
368  if (enb) begin
369  Unit_Delay19_out1 <= i_x3;
370  end
371  end
372  end
373
374
375
376  always @(posedge i_CLK or negedge i_RST_N)
377  begin : Unit_Delay20_process
378  if (i_RST_N == 1'b0) begin
379  Unit_Delay20_out1 <= 8'sb00000000;
380  end
381  else begin
382  if (enb) begin
383  Unit_Delay20_out1 <= i_x0;
384  end
385  end
386  end
387
388
389
390  always @(posedge i_CLK or negedge i_RST_N)
391  begin : Unit_Delay21_process
392  if (i_RST_N == 1'b0) begin
393  Unit_Delay21_out1 <= 8'sb00000000;
394  end
395  else begin
396  if (enb) begin
397  Unit_Delay21_out1 <= i_x1;
398  end
399  end
400  end
401
402
403
404  always @(posedge i_CLK or negedge i_RST_N)
405  begin : Unit_Delay22_process
406  if (i_RST_N == 1'b0) begin
407  Unit_Delay22_out1 <= 8'sb00000000;
408  end
409  else begin
410  if (enb) begin
411  Unit_Delay22_out1 <= i_x2;
412  end
413  end
414  end
415
416
417
418  always @(posedge i_CLK or negedge i_RST_N)
419  begin : Unit_Delay23_process
420  if (i_RST_N == 1'b0) begin
421  Unit_Delay23_out1 <= 8'sb00000000;
422  end
423  else begin
424  if (enb) begin
425  Unit_Delay23_out1 <= i_x3;
426  end
427  end
428  end
429
430
431
432  always @(posedge i_CLK or negedge i_RST_N)
433  begin : Unit_Delay24_process
434  if (i_RST_N == 1'b0) begin
435  Unit_Delay24_out1 <= 8'sb00000000;
436  end
437  else begin
438  if (enb) begin
439  Unit_Delay24_out1 <= i_x0;
440  end
441  end
442  end
443
444
445
446  always @(posedge i_CLK or negedge i_RST_N)
447  begin : Unit_Delay25_process
448  if (i_RST_N == 1'b0) begin
449  Unit_Delay25_out1 <= 8'sb00000000;
450  end
451  else begin
452  if (enb) begin
453  Unit_Delay25_out1 <= i_x1;
454  end
455  end
456  end
457
458
459
460  always @(posedge i_CLK or negedge i_RST_N)
461  begin : Unit_Delay26_process
462  if (i_RST_N == 1'b0) begin
463  Unit_Delay26_out1 <= 8'sb00000000;
464  end
465  else begin
466  if (enb) begin
467  Unit_Delay26_out1 <= i_x2;
468  end
469  end
470  end
471
472
473
474  always @(posedge i_CLK or negedge i_RST_N)
475  begin : Unit_Delay27_process
476  if (i_RST_N == 1'b0) begin
477  Unit_Delay27_out1 <= 8'sb00000000;
478  end
479  else begin
480  if (enb) begin
481  Unit_Delay27_out1 <= i_x3;
482  end
483  end
484  end
485
486
487
488  always @(posedge i_CLK or negedge i_RST_N)
489  begin : Unit_Delay28_process
490  if (i_RST_N == 1'b0) begin
491  Unit_Delay28_out1 <= 8'sb00000000;
492  end
493  else begin
494  if (enb) begin
495  Unit_Delay28_out1 <= i_x0;
496  end
497  end
498  end
499
500
501
502  always @(posedge i_CLK or negedge i_RST_N)
503  begin : Unit_Delay29_process
504  if (i_RST_N == 1'b0) begin
505  Unit_Delay29_out1 <= 8'sb00000000;
506  end
507  else begin
508  if (enb) begin
509  Unit_Delay29_out1 <= i_x1;
510  end
511  end
512  end
513
514
515
516  always @(posedge i_CLK or negedge i_RST_N)
517  begin : Unit_Delay30_process
518  if (i_RST_N == 1'b0) begin
519  Unit_Delay30_out1 <= 8'sb00000000;
520  end
521  else begin
522  if (enb) begin
523  Unit_Delay30_out1 <= i_x2;
524  end
525  end
526  end
527
528
529
530  always @(posedge i_CLK or negedge i_RST_N)
531  begin : Unit_Delay31_process
532  if (i_RST_N == 1'b0) begin
533  Unit_Delay31_out1 <= 8'sb00000000;
534  end
535  else begin
536  if (enb) begin
537  Unit_Delay31_out1 <= i_x3;
538  end
539  end
540  end
541
542
543
544  always @(posedge i_CLK or negedge i_RST_N)
545  begin : Unit_Delay32_process
546  if (i_RST_N == 1'b0) begin
547  Unit_Delay32_out1 <= 8'sb00000000;
548  end
549  else begin
550  if (enb) begin
551  Unit_Delay32_out1 <= i_x0;
552  end
553  end
554  end
555
556
557
558  always @(posedge i_CLK or negedge i_RST_N)
559  begin : Unit_Delay33_process
560  if (i_RST_N == 1'b0) begin
561  Unit_Delay33_out1 <= 8'sb00000000;
562  end
563  else begin
564  if (enb) begin
565  Unit_Delay33_out1 <= i_x1;
566  end
567  end
568  end
569
570
571
572  always @(posedge i_CLK or negedge i_RST_N)
573  begin : Unit_Delay34_process
574  if (i_RST_N == 1'b0) begin
575  Unit_Delay34_out1 <= 8'sb00000000;
576  end
577  else begin
578  if (enb) begin
579  Unit_Delay34_out1 <= i_x2;
580  end
581  end
582  end
583
584
585
586  always @(posedge i_CLK or negedge i_RST_N)
587  begin : Unit_Delay35_process
588  if (i_RST_N == 1'b0) begin
589  Unit_Delay35_out1 <= 8'sb00000000;
590  end
591  else begin
592  if (enb) begin
593  Unit_Delay35_out1 <= i_x3;
594  end
595  end
596  end
597
598
599
600  always @(posedge i_CLK or negedge i_RST_N)
601  begin : Unit_Delay36_process
602  if (i_RST_N == 1'b0) begin
603  Unit_Delay36_out1 <= 8'sb00000000;
604  end
605  else begin
606  if (enb) begin
607  Unit_Delay36_out1 <= i_x0;
608  end
609  end
610  end
611
612
613
614  always @(posedge i_CLK or negedge i_RST_N)
615  begin : Unit_Delay37_process
616  if (i_RST_N == 1'b0) begin
617  Unit_Delay37_out1 <= 8'sb00000000;
618  end
619  else begin
620  if (enb) begin
621  Unit_Delay37_out1 <= i_x1;
622  end
623  end
624  end
625
626
627
628  always @(posedge i_CLK or negedge i_RST_N)
629  begin : Unit_Delay38_process
630  if (i_RST_N == 1'b0) begin
631  Unit_Delay38_out1 <= 8'sb00000000;
632  end
633  else begin
634  if (enb) begin
635  Unit_Delay38_out1 <= i_x2;
636  end
637  end
638  end
639
640
641
642  always @(posedge i_CLK or negedge i_RST_N)
643  begin : Unit_Delay39_process
644  if (i_RST_N == 1'b0) begin
645  Unit_Delay39_out1 <= 8'sb00000000;
646  end
647  else begin
648  if (enb) begin
649  Unit_Delay39_out1 <= i_x3;
650  end
651  end
652  end
653
654
655
656  always @(posedge i_CLK or negedge i_RST_N)
657  begin : Unit_Delay40_process
658  if (i_RST_N == 1'b0) begin
659  Unit_Delay40_out1 <= 8'sb00000000;
660  end
661  else begin
662  if (enb) begin
663  Unit_Delay40_out1 <= i_x0;
664  end
665  end
666  end
667
668
669
670  always @(posedge i_CLK or negedge i_RST_N)
671  begin : Unit_Delay41_process
672  if (i_RST_N == 1'b0) begin
673  Unit_Delay41_out1 <= 8'sb00000000;
674  end
675  else begin
676  if (enb) begin
677  Unit_Delay41_out1 <= i_x1;
678  end
679  end
680  end
681
682
683
684  always @(posedge i_CLK or negedge i_RST_N)
685  begin : Unit_Delay42_process
686  if (i_RST_N == 1'b0) begin
687  Unit_Delay42_out1 <= 8'sb00000000;
688  end
689  else begin
690  if (enb) begin
691  Unit_Delay42_out1 <= i_x2;
692  end
693  end
694  end
695
696
697
698  always @(posedge i_CLK or negedge i_RST_N)
699  begin : Unit_Delay43_process
700  if (i_RST_N == 1'b0) begin
701  Unit_Delay43_out1 <= 8'sb00000000;
702  end
703  else begin
704  if (enb) begin
705  Unit_Delay43_out1 <= i_x3;
706  end
707  end
708  end
709
710
711
712  always @(posedge i_CLK or negedge i_RST_N)
713  begin : Unit_Delay44_process
714  if (i_RST_N == 1'b0) begin
715  Unit_Delay44_out1 <= 8'sb00000000;
716  end
717  else begin
718  if (enb) begin
719  Unit_Delay44_out1 <= i_x0;
720  end
721  end
722  end
723
724
725
726  always @(posedge i_CLK or negedge i_RST_N)
727  begin : Unit_Delay45_process
728  if (i_RST_N == 1'b0) begin
729  Unit_Delay45_out1 <= 8'sb00000000;
730  end
731  else begin
732  if (enb) begin
733  Unit_Delay45_out1 <= i_x1;
734  end
735  end
736  end
737
738
739
740  always @(posedge i_CLK or negedge i_RST_N)
741  begin : Unit_Delay46_process
742  if (i_RST_N == 1'b0) begin
743  Unit_Delay46_out1 <= 8'sb00000000;
744  end
745  else begin
746  if (enb) begin
747  Unit_Delay46_out1 <= i_x2;
748  end
749  end
750  end
751
752
753
754  always @(posedge i_CLK or negedge i_RST_N)
755  begin : Unit_Delay47_process
756  if (i_RST_N == 1'b0) begin
757  Unit_Delay47_out1 <= 8'sb00000000;
758  end
759  else begin
760  if (enb) begin
761  Unit_Delay47_out1 <= i_x3;
762  end
763  end
764  end
765
766
767
768  always @(posedge i_CLK or negedge i_RST_N)
769  begin : Unit_Delay48_process
770  if (i_RST_N == 1'b0) begin
771  Unit_Delay48_out1 <= 8'sb00000000;
772  end
773  else begin
774  if (enb) begin
775  Unit_Delay48_out1 <= i_x0;
776  end
777  end
778  end
779
780
781
782  always @(posedge i_CLK or negedge i_RST_N)
783  begin : Unit_Delay49_process
784  if (i_RST_N == 1'b0) begin
785  Unit_Delay49_out1 <= 8'sb00000000;
786  end
787  else begin
788  if (enb) begin
789  Unit_Delay49_out1 <= i_x1;
790  end
791  end
792  end
793
794
795
796  always @(posedge i_CLK or negedge i_RST_N)
797  begin : Unit_Delay50_process
798  if (i_RST_N == 1'b0) begin
799  Unit_Delay50_out1 <= 8'sb00000000;
800  end
801  else begin
802  if (enb) begin
803  Unit_Delay50_out1 <= i_x2;
804  end
805  end
806  end
807
808
809
810  always @(posedge i_CLK or negedge i_RST_N)
811  begin : Unit_Delay51_process
812  if (i_RST_N == 1'b0) begin
813  Unit_Delay51_out1 <= 8'sb00000000;
814  end
815  else begin
816  if (enb) begin
817  Unit_Delay51_out1 <= i_x3;
818  end
819  end
820  end
821
822
823
824  always @(posedge i_CLK or negedge i_RST_N)
825  begin : Unit_Delay52_process
826  if (i_RST_N == 1'b0) begin
827  Unit_Delay52_out1 <= 8'sb00000000;
828  end
829  else begin
830  if (enb) begin
831  Unit_Delay52_out1 <= i_x0;
832  end
833  end
834  end
835
836
837
838  always @(posedge i_CLK or negedge i_RST_N)
839  begin : Unit_Delay53_process
840  if (i_RST_N == 1'b0) begin
841  Unit_Delay53_out1 <= 8'sb00000000;
842  end
843  else begin
844  if (enb) begin
845  Unit_Delay53_out1 <= i_x1;
846  end
847  end
848  end
849
850
851
852  always @(posedge i_CLK or negedge i_RST_N)
853  begin : Unit_Delay54_process
854  if (i_RST_N == 1'b0) begin
855  Unit_Delay54_out1 <= 8'sb00000000;
856  end
857  else begin
858  if (enb) begin
859  Unit_Delay54_out1 <= i_x2;
860  end
861  end
862  end
863
864
865
866  always @(posedge i_CLK or negedge i_RST_N)
867  begin : Unit_Delay55_process
868  if (i_RST_N == 1'b0) begin
869  Unit_Delay55_out1 <= 8'sb00000000;
870  end
871  else begin
872  if (enb) begin
873  Unit_Delay55_out1 <= i_x3;
874  end
875  end
876  end
877
878
879
880  always @(posedge i_CLK or negedge i_RST_N)
881  begin : Unit_Delay56_process
882  if (i_RST_N == 1'b0) begin
883  Unit_Delay56_out1 <= 8'sb00000000;
884  end
885  else begin
886  if (enb) begin
887  Unit_Delay56_out1 <= i_x0;
888  end
889  end
890  end
891
892
893
894  always @(posedge i_CLK or negedge i_RST_N)
895  begin : Unit_Delay57_process
896  if (i_RST_N == 1'b0) begin
897  Unit_Delay57_out1 <= 8'sb00000000;
898  end
899  else begin
900  if (enb) begin
901  Unit_Delay57_out1 <= i_x1;
902  end
903  end
904  end
905
906
907
908  always @(posedge i_CLK or negedge i_RST_N)
909  begin : Unit_Delay58_process
910  if (i_RST_N == 1'b0) begin
911  Unit_Delay58_out1 <= 8'sb00000000;
912  end
913  else begin
914  if (enb) begin
915  Unit_Delay58_out1 <= i_x2;
916  end
917  end
918  end
919
920
921
922  always @(posedge i_CLK or negedge i_RST_N)
923  begin : Unit_Delay59_process
924  if (i_RST_N == 1'b0) begin
925  Unit_Delay59_out1 <= 8'sb00000000;
926  end
927  else begin
928  if (enb) begin
929  Unit_Delay59_out1 <= i_x3;
930  end
931  end
932  end
933
934
935
936  always @(posedge i_CLK or negedge i_RST_N)
937  begin : Unit_Delay60_process
938  if (i_RST_N == 1'b0) begin
939  Unit_Delay60_out1 <= 8'sb00000000;
940  end
941  else begin
942  if (enb) begin
943  Unit_Delay60_out1 <= i_x0;
944  end
945  end
946  end
947
948
949
950  always @(posedge i_CLK or negedge i_RST_N)
951  begin : Unit_Delay61_process
952  if (i_RST_N == 1'b0) begin
953  Unit_Delay61_out1 <= 8'sb00000000;
954  end
955  else begin
956  if (enb) begin
957  Unit_Delay61_out1 <= i_x1;
958  end
959  end
960  end
961
962
963
964  always @(posedge i_CLK or negedge i_RST_N)
965  begin : Unit_Delay62_process
966  if (i_RST_N == 1'b0) begin
967  Unit_Delay62_out1 <= 8'sb00000000;
968  end
969  else begin
970  if (enb) begin
971  Unit_Delay62_out1 <= i_x2;
972  end
973  end
974  end
975
976
977
978  always @(posedge i_CLK or negedge i_RST_N)
979  begin : Unit_Delay63_process
980  if (i_RST_N == 1'b0) begin
981  Unit_Delay63_out1 <= 8'sb00000000;
982  end
983  else begin
984  if (enb) begin
985  Unit_Delay63_out1 <= i_x3;
986  end
987  end
988  end
989
990
991
992  always @(posedge i_CLK or negedge i_RST_N)
993  begin : Unit_Delay64_process
994  if (i_RST_N == 1'b0) begin
995  Unit_Delay64_out1 <= 8'sb00000000;
996  end
997  else begin
998  if (enb) begin
999  Unit_Delay64_out1 <= i_x0;
1000  end
1001  end
1002  end
1003
1004
1005
1006  always @(posedge i_CLK or negedge i_RST_N)
1007  begin : Unit_Delay65_process
1008  if (i_RST_N == 1'b0) begin
1009  Unit_Delay65_out1 <= 8'sb00000000;
1010  end
1011  else begin
1012  if (enb) begin
1013  Unit_Delay65_out1 <= i_x1;
1014  end
1015  end
1016  end
1017
1018
1019
1020  always @(posedge i_CLK or negedge i_RST_N)
1021  begin : Unit_Delay66_process
1022  if (i_RST_N == 1'b0) begin
1023  Unit_Delay66_out1 <= 8'sb00000000;
1024  end
1025  else begin
1026  if (enb) begin
1027  Unit_Delay66_out1 <= i_x2;
1028  end
1029  end
1030  end
1031
1032
1033
1034  always @(posedge i_CLK or negedge i_RST_N)
1035  begin : Unit_Delay67_process
1036  if (i_RST_N == 1'b0) begin
1037  Unit_Delay67_out1 <= 8'sb00000000;
1038  end
1039  else begin
1040  if (enb) begin
1041  Unit_Delay67_out1 <= i_x3;
1042  end
1043  end
1044  end
1045
1046
1047
1048  always @(posedge i_CLK or negedge i_RST_N)
1049  begin : Unit_Delay68_process
1050  if (i_RST_N == 1'b0) begin
1051  Unit_Delay68_out1 <= 8'sb00000000;
1052  end
1053  else begin
1054  if (enb) begin
1055  Unit_Delay68_out1 <= i_x0;
1056  end
1057  end
1058  end
1059
1060
1061
1062  always @(posedge i_CLK or negedge i_RST_N)
1063  begin : Unit_Delay69_process
1064  if (i_RST_N == 1'b0) begin
1065  Unit_Delay69_out1 <= 8'sb00000000;
1066  end
1067  else begin
1068  if (enb) begin
1069  Unit_Delay69_out1 <= i_x1;
1070  end
1071  end
1072  end
1073
1074
1075
1076  always @(posedge i_CLK or negedge i_RST_N)
1077  begin : Unit_Delay70_process
1078  if (i_RST_N == 1'b0) begin
1079  Unit_Delay70_out1 <= 8'sb00000000;
1080  end
1081  else begin
1082  if (enb) begin
1083  Unit_Delay70_out1 <= i_x2;
1084  end
1085  end
1086  end
1087
1088
1089
1090  always @(posedge i_CLK or negedge i_RST_N)
1091  begin : Unit_Delay71_process
1092  if (i_RST_N == 1'b0) begin
1093  Unit_Delay71_out1 <= 8'sb00000000;
1094  end
1095  else begin
1096  if (enb) begin
1097  Unit_Delay71_out1 <= i_x3;
1098  end
1099  end
1100  end
1101
1102
1103
1104  always @(posedge i_CLK or negedge i_RST_N)
1105  begin : Unit_Delay72_process
1106  if (i_RST_N == 1'b0) begin
1107  Unit_Delay72_out1 <= 8'sb00000000;
1108  end
1109  else begin
1110  if (enb) begin
1111  Unit_Delay72_out1 <= i_x0;
1112  end
1113  end
1114  end
1115
1116
1117
1118  always @(posedge i_CLK or negedge i_RST_N)
1119  begin : Unit_Delay73_process
1120  if (i_RST_N == 1'b0) begin
1121  Unit_Delay73_out1 <= 8'sb00000000;
1122  end
1123  else begin
1124  if (enb) begin
1125  Unit_Delay73_out1 <= i_x1;
1126  end
1127  end
1128  end
1129
1130
1131
1132  always @(posedge i_CLK or negedge i_RST_N)
1133  begin : Unit_Delay74_process
1134  if (i_RST_N == 1'b0) begin
1135  Unit_Delay74_out1 <= 8'sb00000000;
1136  end
1137  else begin
1138  if (enb) begin
1139  Unit_Delay74_out1 <= i_x2;
1140  end
1141  end
1142  end
1143
1144
1145
1146  always @(posedge i_CLK or negedge i_RST_N)
1147  begin : Unit_Delay75_process
1148  if (i_RST_N == 1'b0) begin
1149  Unit_Delay75_out1 <= 8'sb00000000;
1150  end
1151  else begin
1152  if (enb) begin
1153  Unit_Delay75_out1 <= i_x3;
1154  end
1155  end
1156  end
1157
1158
1159
1160  always @(posedge i_CLK or negedge i_RST_N)
1161  begin : Unit_Delay76_process
1162  if (i_RST_N == 1'b0) begin
1163  Unit_Delay76_out1 <= 8'sb00000000;
1164  end
1165  else begin
1166  if (enb) begin
1167  Unit_Delay76_out1 <= i_x0;
1168  end
1169  end
1170  end
1171
1172
1173
1174  always @(posedge i_CLK or negedge i_RST_N)
1175  begin : Unit_Delay77_process
1176  if (i_RST_N == 1'b0) begin
1177  Unit_Delay77_out1 <= 8'sb
```

```
148      eriu
149
150
151
152      always @(posedge i_CLK or negedge i_RST_N)
153      begin : Unit_Delay4_process
154      if (i_RST_N == 1'b0) begin
155      Unit_Delay4_out1 <= 8'sb00000000;
156      end
157      else begin
158      if (enb) begin
159      Unit_Delay4_out1 <= i_X0;
160      end
161      end
162      end
163
164
165
166      assign Product_out1 = Unit_Delay_out1 * Unit_Delay4_out1;
167
168
169
170      always @(posedge i_CLK or negedge i_RST_N)
171      begin : Unit_Delay5_process
172      if (i_RST_N == 1'b0) begin
173      Unit_Delay5_out1 <= 8'sb00000000;
174      end
175      else begin
176      if (enb) begin
177      Unit_Delay5_out1 <= i_X1;
178      end
179      end
180      end
181
182
183
184      assign Product1_out1 = Unit_Delay1_out1 * Unit_Delay5_out1;
185
186
187
188      assign Add_stage2_add_temp = Product_out1 + Product1_out1;
189      assign Add_op_stage1 = {Add_stage2_add_temp[15], Add_stage2_add_temp};
190
191
192
193      always @(posedge i_CLK or negedge i_RST_N)
194      begin : Unit_Delay6_process
195      if (i_RST_N == 1'b0) begin
196      Unit_Delay6_out1 <= 8'sb00000000;
197      end
198      else begin
199      if (enb) begin
200      Unit_Delay6_out1 <= i_X2;
201      end
202      end
203      end
```

```
199      if (enb) begin
200      Unit_Delay6_out1 <= i_X2;
201      end
202      end
203      end
204
205
206
207      assign Product2_out1 = Unit_Delay2_out1 * Unit_Delay6_out1;
208
209
210
211      assign Add_stage3_add_cast = Add_op_stage1[15:0];
212      assign Add_stage3_add_temp = Add_stage3_add_cast + Product2_out1;
213      assign Add_op_stage2 = {{2{Add_stage3_add_temp[15]}}}, Add_stage3_add_temp};
214
215
216
217      always @(posedge i_CLK or negedge i_RST_N)
218      begin : Unit_Delay7_process
219      if (i_RST_N == 1'b0) begin
220      Unit_Delay7_out1 <= 8'sb00000000;
221      end
222      else begin
223      if (enb) begin
224      Unit_Delay7_out1 <= i_X3;
225      end
226      end
227      end
228
229
230
231      assign Product3_out1 = Unit_Delay3_out1 * Unit_Delay7_out1;
232
233
234
235      assign Add_stage4_add_cast = Add_op_stage2[15:0];
236      assign Add_out1 = Add_stage4_add_cast + Product3_out1;
237
238
239
240      always @(posedge i_CLK or negedge i_RST_N)
241      begin : Unit_Delay8_process
242      if (i_RST_N == 1'b0) begin
243      Unit_Delay8_out1 <= 16'sb0000000000000000;
244      end
245      else begin
246      if (enb) begin
247      Unit_Delay8_out1 <= Add_out1;
248      end
249      end
250      end
251
252
253
```

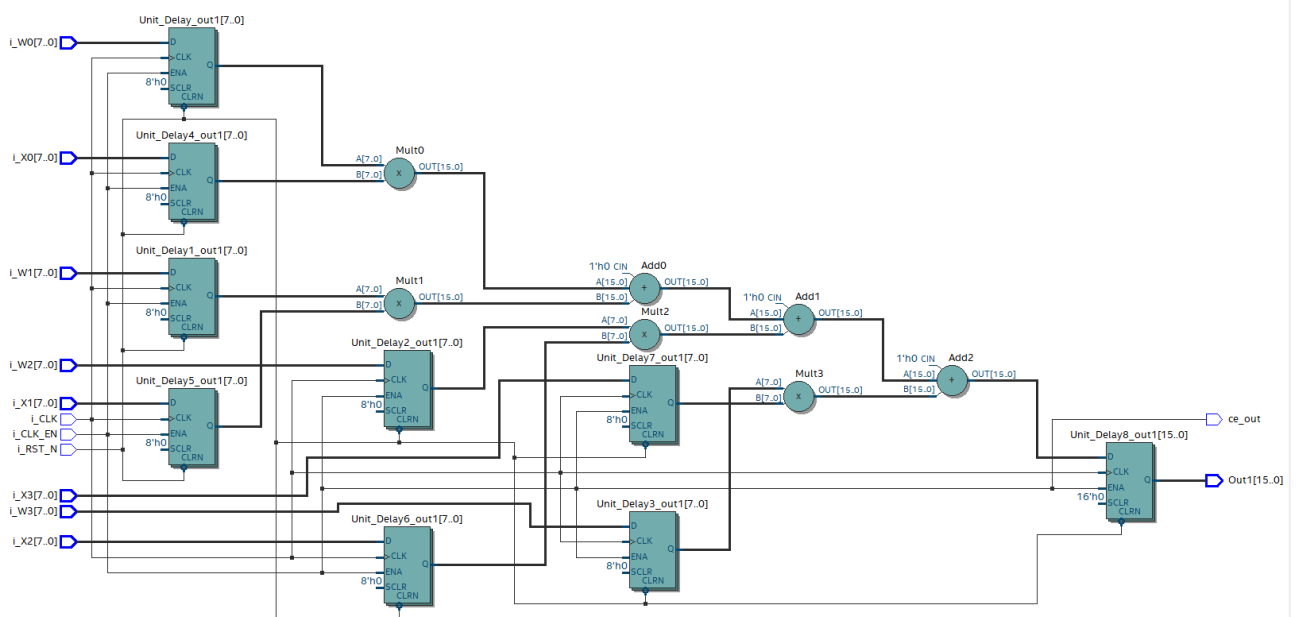


```

235 assign Add_stage4_add_cast = Add_op_stage2[15:0];
236 assign Add_out1 = Add_stage4_add_cast + Product3_out1;
237
238
239
240 always @(posedge i_CLK or negedge i_RST_N)
241   begin : Unit_Delay8_process
242     if (i_RST_N == 1'b0) begin
243       Unit_Delay8_out1 <= 16'sb0000000000000000;
244     end
245     else begin
246       if (enb) begin
247         Unit_Delay8_out1 <= Add_out1;
248       end
249     end
250   end
251
252
253
254 assign Out1 = Unit_Delay8_out1;
255 assign ce_out = i_CLK_EN;
256
257 endmodule // Subsystem

```

Результат в RTL Viewer:



Висновок: під час виконання цієї лабораторної роботи я розробив підсистему і перевіряв її роботу. Можна сказати, що симуляція в програмних забезпеченнях Matlab та синтез в Quartus мають задовільний результат.