FBU-DAE 2021
1 (1) : 54-68

NEURAL NETWORK BASED PERFORMANCE ESTIMATION METHODOLOGY OF FPGA FPU IP'S FOR THE DESIGNS WITH HIGH PERFORMANCE REQUIREMENTS

YÜKSEK PERFORMANS GEREKLİLİKLERİNE SAHİP TASARIMLAR İÇİN FPGA FPU IP'LERİNİN SİNİR AĞ TABANLI PERFORMANS TAHMİN METODOLOJİSİ

Vecdi Emre LEVENT *

ABSTRACT

In applications requiring high precision calculations, floating point number representation is preferred instead of fixed point number representation system. The main reason is that floating point number representation can express numbers in a much wider range. Since the design of floating point arithmetic processing units is a difficult process, the use of floating point units provided by FPGA design companies (such as Xilinx, Intel) can be preferred for arithmetic operations involving floating point in an algorithm design process on an FPGA. In case the use of IPs offered by FPGA manufacturers is preferred, the area usage and maximum frequency parameters of these IPs are not predicted before design. This situation raises the need to obtain floating point units that give the maximum frequency with the minimum area, especially when there is a requirement to maximize the output of the system. However, it is necessary to wait for minutes depending on the computing power in order to obtain the frequency and occupied area for a simple floating point unit even with a fixed latency setting. In this study, in case the output of a system needs to be maximum, a methodology is given to predict the highest performing floating point units before synthesizing them. It has been shown that the correct FPU selection can be made before the design process with the proposed batch synthesis tool and artificial neural network approach.

Keywords: FPGA, FPU, Estimation, Neural Network, Throughput

ÖZET

Yüksek hassasiyet ile hesaplama yapılması gereken uygulamalarda fixed point sayı gösterim sistemi yerine floating point sayı gösterimi tercih edilmektedir. Bunun başlıca nedeni floating point sayı gösteriminin çok daha geniş aralıkta sayıları ifade edebiliyor olmasıdır. Floating point aritmetik işlem ünitelerinin tasarımı zor bir süreç olduğundan ötürü, FPGA üzerinde bir algoritma tasarım sürecinde floating point içeren aritmetik işlemler için, FPGA tasarım firmalarının (Xilinx, Intel gibi) sağladığı floating point ünitelerin kullanımı tercih edilebilmektedir. FPGA üretici firmalarının sunmuş olduğu IP'lerin kullanımı tercih edildiği durumda ise bu IP'lerin alan kullanımı, çıkabilecekleri maksimum frekans parametrelerinin tasarımdan önce tahmin edilmemesidir. Bu durum özellikle sistemin çıktısını maksimum olacak şekilde bir gereksinim olduğunda, minimum alan ile maksimum frekansı veren floating point ünitelerinin elde edilme ihtiyacını doğurmaktadır. Ancak en basit bir floating point ünitesinin sabit bir gecikme değeri ile bile ne kadar alan kaplayacağı ve çıkabileceği frekans değerini elde etmek için işlem gücüne bağlı olarak dakikalarca beklemek gerekmektedir. Bu çalışmada, bir sistemin çıktısının maksimum olma ihtiyacı durumunda, floating point ünitelerinin en yüksek performanslı olanlarını daha sentez yapmadan öngörecek bir metodoloji verilmektedir. Bahsedilen toplu sentez aracı ve yapay sinir ağları yaklaşımı ile doğru FPU seçiminin tasarıma başlanmadan yapılabildiği gösterilmiştir.

Anahtar Kelimeler: FPGA, FPU, Yapay Sinir Ağları, Tahmin Etme, Veri Çıktısı

*
Bilgisayar Mühendisliği Bölümü,
Fenerbahçe Üniversitesi, İstanbul / Türkiye

Department of Computer Engineering,
Fenerbahçe University, Istanbul / Turkey

ORCID: 0000-0001-6886-8875

1. INTRODUCTION

Nowadays, FPGA designers grapple with high performance requirements in almost every FPGA design process. These requirements often greatly increase the development time of the design. If the algorithm to be designed includes a floating point operation, there are two ways to develop this operation on FPGA. The first way is to develop your own custom solution for the relevant floating point operation. The other option is to prefer to use IP based Floating Point Units, which are provided by FPGA companies. Since the design process of floating point units is very challenging, designers often turn to IP-based solutions. However, in cases where maximum performance is expected as a design goal; For example, when the highest FPS value obtainable in a video processing application is desired; Ready-made IP-based solutions have one drawback. FPU IPs of the two most prominent companies in the market, Xilinx and Intel, are produced according to the latency parameter. Choosing

too little or too much latency affects the performance of the design it will produce. This performance can be evaluated as minimum area and maximum frequency. At this point, it may be necessary to do thousands of syntheses with different latency values to determine the optimum point. This means a time requirement of days or even months for development. The time elapsed when the design is made without this approach has been examined in various publications (Ugurdag, 2013), (Levent, 2020), (Levent, 2018).

There are studies in the literature to predict resource use before starting design (Schumacher, 2018), (Bjureus, 2002), (Degalahal, 2005), (Milder, 2006), (Shi, 2004). The main purpose of these studies is to inform the designer about the design that will emerge before starting the design. This information is very useful for making design decisions ahead of time.

In order to reduce development time while maximizing performance, a methodology is given with the batch synthesis approach and modeling approach with artificial neural networks. With this approach, the determination of the point where the design will give its maximum performance can be reduced to seconds. To test this methodology, Arria 10 FPGAs, which Intel has released as the last generation (Intel, 2021), have been used and it has been shown that the methodology can dramatically reduce the development time.

2. BATCH SYTHESIS METHODOLOGY

The main aim of the study is to provide approaches for estimating the area and frequency values of FPU units without repetitive synthesis each time. The main purpose in batch synthesis methodology is to automate the acquisition of the Floating Point Units synthesis results of the target FPGA family.

The following steps are made to automate this process.

1. A project is created for the selected floating point unit of the FPGA manufacturer used.
2. Project copies are produced with different values from 1 to 30 of the latecy values of the FPU in this project.
3. Synthesising by creating parallel threads according to the CPU core on the server.
4. The area and frequency results are obtained from the reports provided by the FPGA manufacturer synthesis tool (e.g. Xilinx Vivado).

Figure 1 shows the batch synthesis methodology.

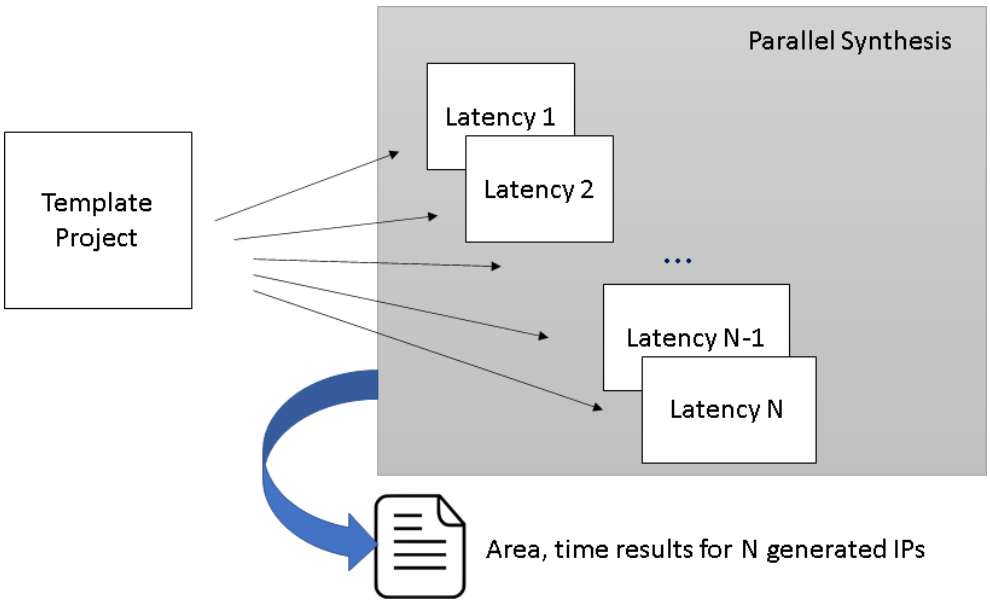


Figure 1. Batch Syntehsis Methodolgy

3. NEURAL NETWORK BASED ESTIMATION

Artificial Neural Networks (ANNs), inspired by the information processing method in the brain, form a network with layers and the neurons within these layers. With the numerical values assigned to the neurons and the transfer functions at the outputs of the layers, calculations are made within the system and the result is obtained from the output layer.

During the training, the effect on the result is examined by changing the values of the neurons with predetermined neuron, layer numbers and transfer functions. The goal is to minimize the error between what should happen in each cycle and the calculated value.

In this study, synthesis results produced by batch syntesis tool in Matlab environment were modeled by artificial neural networks method.

Approximately 30 reports were obtained for each FPU. 70% of the reports obtained from the synthesis tools were used for training and 30% for testing.

Figure 2 gives an example for artificial neural networks used in the Matlab environment.

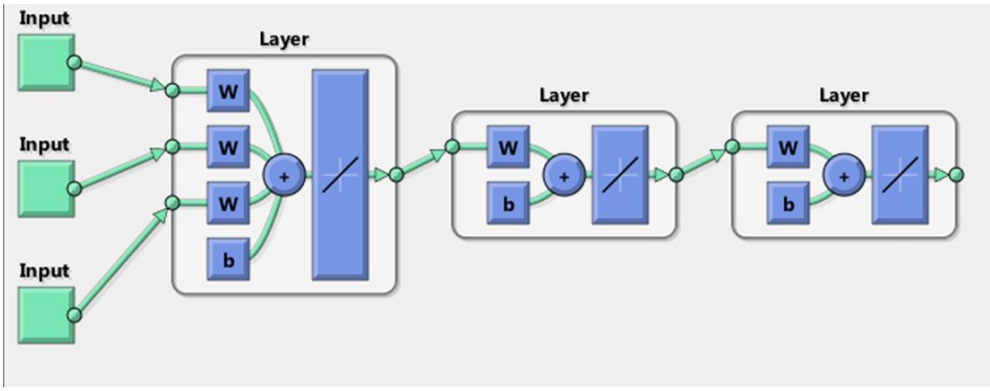


Figure 2. Batch Syntehsis Methodolgy

4. A USE CASE ON ARRIA 10 FPGAS

Under this heading, an example is given that the FPU analysis of the new generation FPGAs released by Intel can be done with the methodologies mentioned in the article. Synthesis results of Arria 10 FPGAs will be given in subtitles.

Arria 10 FPGAs are developed with 20nm manufacturing technology. It offers low power consumption and high performance. General features of Arria 10 GX 1150 FPGA are given below.

- Input/Output Pins: 600
- Registes: 1708800
- Manufacturing Technology 20nm
- RAM Bits 54260
- Logic Cells 1150000
- Fixed-Point Multipliers 3036 (18x19)
- Embedded Floating-Point Units 1518

It works with a structure that other companies have not yet done. Altera has embedded addition and multiplication units that perform floating-point processing in FPGA. It has the capability of processing at gigaherz levels in addition and multiplication operations. Although there are no embedded units in other arithmetic operations, it is possible to reach +800 MHz speeds with deep pipeline structures, as the FPGA is produced with the 20nm standard.

4.1 Floating-Point Unit Analysis

More floating-point processing units are supported in Arria 10 compared to previous generation FPGAs. These are given in Table 1.

Arithmetic	Converters	Comparators	Exponential	Trigonometric
Add	Fixed -> Float	Minimum	Exponential (2, 10, e)	Sin
Sub	Float -> Fixed	Maximum	Log (2, 10, e)	Cos
Add/Sub	Bit width convert	Smaller or Bigger	Log (1+x)	Tan
Multiply			LDExp	Arcsin
Division				Arccos
Absolute Value				Arctan
Scalar Product				
Root				
Square Root				
Cube Root				
Inverse Square				

Table 1. Supported Floating Point Units

FPUs embedded in Arria 10 can be used for addition, subtraction and addition / subtraction units in arithmetic operations. Embedded FPUs have the advantage of having an embedded circuit for the associated operation. Therefore, it can operate at high frequencies. The analysis of the FPUs used in the project will be given in the subtitles.

4.1.1 FPU Adder Unit

The adder unit can be used in two ways. The first is to use embedded FPUs and the other method is to create using LUTs.

The embedded FPU is used by feeding data to the FPU's input register and taking it after 3 cycles. Relevant values are given in Table 2.

ALM	25
Register	96
DSP Block	1
Frequency(MHz)	1021
Latency	3

Table 2. Adder Embedded FPU Area-Frequency Specs

The internal structure of the embedded FPU is given in Figure 3.

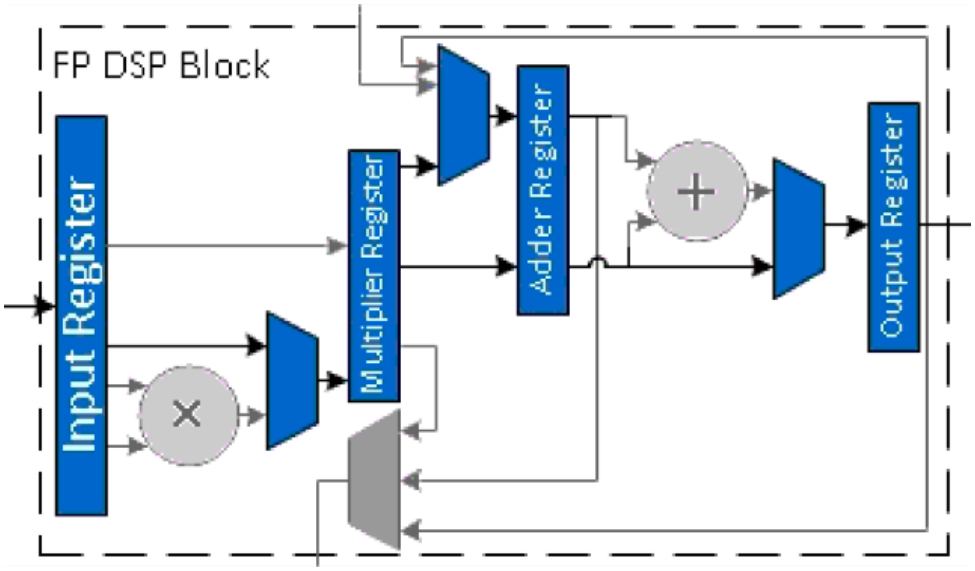


Figure 3. Arria 10 Embedded FPUs

When the adder unit is created using LUTs, the performance varies according to the selected delay. In order to select the most efficient FPU, a metric called Area-Time product was created. The number of ALM is calculated as * (1 / Freq) (Area-Time Product metric used in other titles is calculated in the same way). The smallest value is the best result. The best result is shown in green. Results are given in Table 3.

Figure 4 shows the performance effect graphs of the Collector FPU according to the delay.

Table 3. LUT Based Adder FPU Synthesis Results

Latency	ALM	Register	Freq(MHz)	Area-Time Product
1	328	169	146,95	2,232051718
2	320	224	203,21	1,574725653
3	345	299	234,85	1,46902278
4	348	313	268,24	1,297345661
5	315	409	339,67	0,927370683
6	465	347	386,1	1,204351204
7	304	542	418,24	0,726855394
8	305	576	412,03	0,740237361
9	337	623	412,37	0,817227247
10	342	700	446,83	0,76539176
11	356	758	487,57	0,730151568
12	368	836	543,48	0,677117833
13	373	877	527,98	0,706466154
14	380	881	540,54	0,703000703
15	400	971	504,29	0,793194392
16	411	1028	558,66	0,73568897
17	414	1058	532,2	0,777903044
18	447	1143	629,72	0,709839294
19	462	1224	661,81	0,698085553
20	478	1287	666,67	0,716996415
21	501	1376	672,95	0,744483245
22	507	1438	708,72	0,715374196
23	523	1497	640,61	0,81640936
24	543	1573	662,25	0,81993205
25	551	1609	695,41	0,792338333
26	580	1717	738,01	0,78589721
27	591	1757	800,64	0,738159472
28	608	1660	758,73	0,80133908
29	612	1687	783,7	0,780911063
30	633	2010	758,73	0,834288877

Figure 4. (a) Adder FPU, ALM, Frequency, Register and Area-Time Products Specs

ALM	25
Register	97
DSP Block	1
Frequency(MHz)	985.22
Latency	3

Table 4. Addition / Subtraction Embedded FPU Field-Speed Values

Table 5. Addition / Subtraction FPU Results Made with LUTs

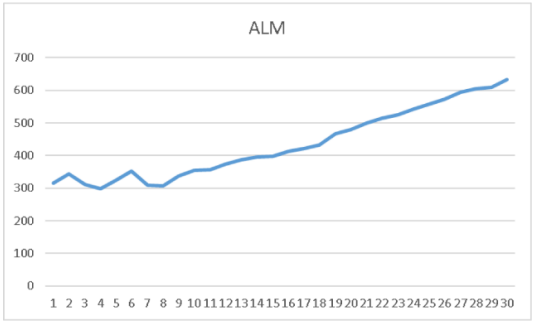
Figure 5. (a) Addition / Subtraction FPU, ALM, Frequency, Register and Area-Time Products Specs

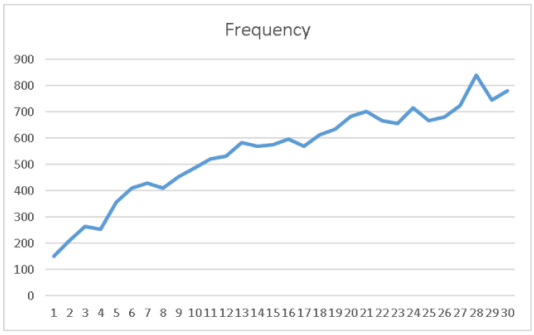
4.1.2 FPU Adder/Subtractor Unit

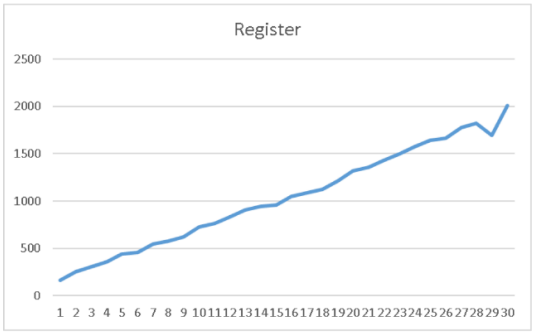
The area and frequency results used by the addition / subtractor of the FPU with embedded FPUs are given in Table 4. The internal structure of the embedded FPU is as in Figure 3 given in the adder section.

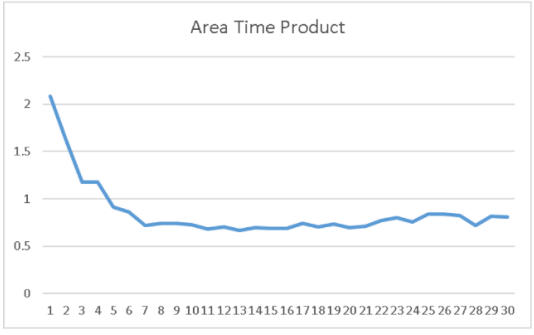
Performance values when Addition / Subtraction FPU is made using LUTs are given in Table 5 according to the delay. The lowest Area-Time Multiplication value is shown in green. Figure 5 shows the performance effect graphs of the Addition / Subtraction FPU according to the delay.

Latecy	ALM	Register	Freq(MHz)	Area-Time Product
1	315	163	151,24	2,082782333
2	344	255	211,06	1,629868284
3	312	310	264,62	1,179049203
4	299	358	253,61	1,178975592
5	324	441	355,24	0,912059453
6	352	460	409,17	0,860278124
7	308	543	428,27	0,719172485
8	306	579	410,68	0,745105678
9	337	623	453,31	0,743420617
10	354	724	485,91	0,728529975
11	356	762	521,65	0,682449919
12	374	833	531,91	0,703126469
13	386	908	582,07	0,66315048
14	395	941	568,83	0,69440782
15	397	963	576,37	0,688793657
16	413	1046	597,37	0,691363811
17	422	1090	569,48	0,741026902
18	433	1125	611,62	0,70795592
19	467	1212	634,52	0,735989409
20	479	1318	683,99	0,700302636
21	498	1361	700,77	0,71064686
22	515	1436	667,56	0,771466235
23	525	1499	655,74	0,800622198
24	542	1572	716,33	0,756634512
25	558	1644	666,67	0,836995815
26	572	1667	679,35	0,841981306
27	593	1774	722,54	0,820715808
28	604	1824	841,04	0,71815847
29	609	1693	745,71	0,81667136
30	632	2012	780,64	0,80959213









ALM	25
Register	96
DSP Block	1
Frequency(MHz)	1112
Latency	3

Table 6. Multiply Embedded FPU Area-Frequency Specs

Table 7. LUT Based Multiply FPU Area-Frequency Specs

Figure 6. (a) Multiply FPU, ALM, Frequency, Register and Area-Time Products Specs

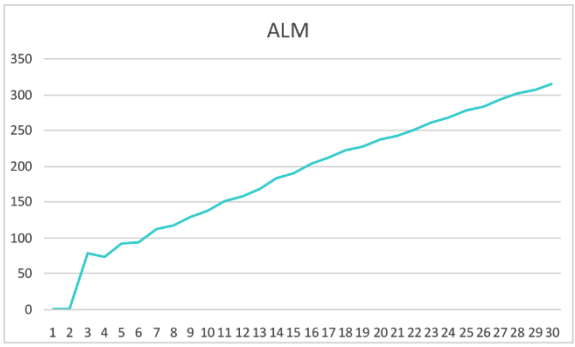
4.1.3 FPU Multipy Unit

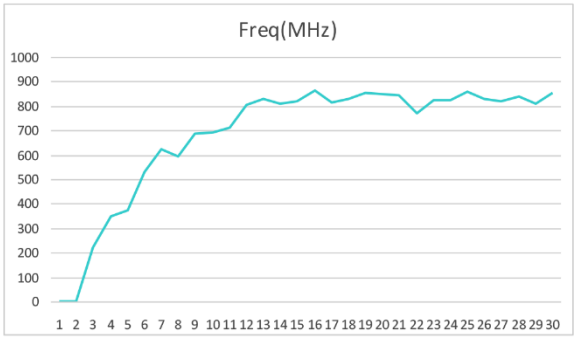
The area and frequency results of the embedded multiplication FPUs are given in Table 6. The internal structure of the embedded FPU is as in Figure 3 given in the adder section.

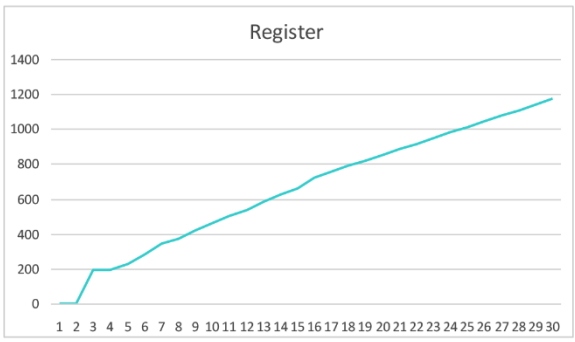
Performance values of the multiplication FPU using LUTs are given in Table 7 according to the delay. The lowest Area-Time Multiplication value is shown in green.

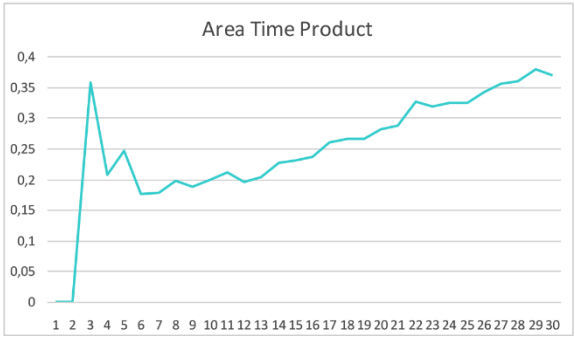
Figure 6 shows the performance effect graphs of the Multiplication FPU according to the delay.

Latency	ALM	Register	Fixed Point DSP	Freq(MHz)	Area-Time Product
1	-	-	-	-	-
2	-	-	-	-	-
3	78	194	1	218,53	0,356930399
4	72	193	1	347,1	0,207433016
5	91	228	1	371,2	0,245150862
6	93	280	1	529,38	0,175677207
7	111	343	1	624,61	0,177710892
8	117	374	1	591,72	0,197728655
9	128	418	1	683,99	0,187137239
10	137	461	1	691,56	0,19810284
11	150	500	1	712,25	0,210600211
12	157	537	1	805,8	0,194837429
13	167	580	1	826,45	0,202069091
14	183	624	1	807,75	0,226555246
15	189	661	1	820,34	0,230392276
16	203	724	1	861,33	0,235682027
17	211	754	1	815,66	0,258686217
18	221	787	1	830,56	0,266085533
19	227	819	1	853,97	0,2658173
20	237	848	1	846,02	0,280135221
21	242	887	1	842,46	0,287253994
22	251	912	1	768,05	0,326801641
23	261	948	1	823,05	0,317113177
24	267	980	1	824,4	0,323871907
25	277	1009	1	856,9	0,323258257
26	283	1041	1	827,13	0,342146942
27	292	1080	1	820,34	0,355949972
28	301	1108	1	838,93	0,358790364
29	306	1138	1	809,72	0,377908413
30	314	1174	1	851,79	0,368635462









4.1.4 FPU Division Unit

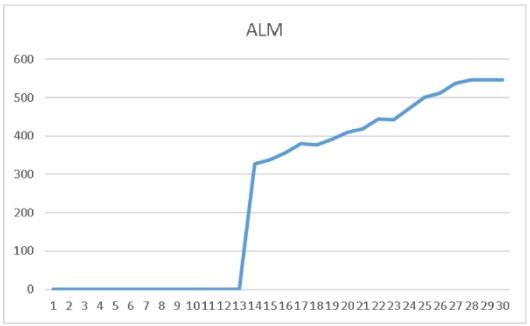
Embedded FPUs cannot be used in division. In addition to ALM and registers, Block RAM and Fixed point processing units are also used. Area performance results are given in Table 8. The line marked with green indicates the highest efficiency FPU.

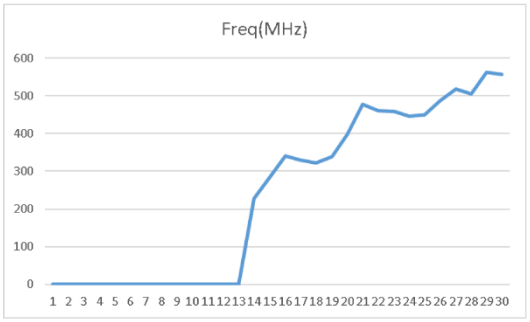
Figure 7 shows the performance effect graphs of the Partition FPU according to the delay.

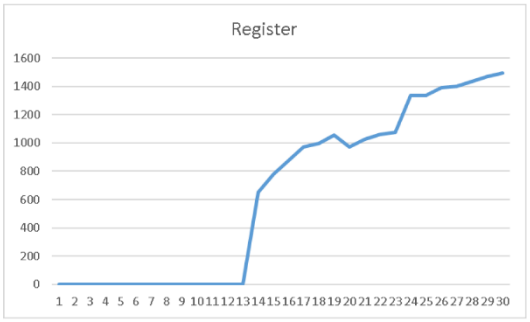
Table 8. Division FPU Area-Frequency Specs

Latency	ALM	Register	Freq (MHz)	Block Memory Bits	RAM Blocks	DSP Blocsk	Area Time Product
1	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-
14	328	654	227.12	33,792	3	4	1.44417
15	338	782	285.63	33,792	3	4	1.183349
16	356	874	340.14	33,792	3	4	1.046628
17	381	971	330.14	33,792	3	4	1.154056
18	377	998	321.54	33,792	3	4	1.172482
19	391	1056	338.87	33,792	3	4	1.153835
20	409	973	397.3	33,792	3	4	1.029449
21	418	1025	478.24	33,792	3	4	0.874038
22	444	1062	461.04	33,792	3	4	0.96304
23	442	1074	458.51	33,792	3	4	0.963992
24	471	1337	446.63	33,792	3	4	1.054564
25	500	1339	449.24	33,792	3	4	1.112991
26	512	1394	487.33	33,792	3	4	1.050623
27	537	1402	517.6	33,792	3	4	1.037481
28	546	1436	505.56	33,792	3	4	1.079991
29	547	1469	562.11	33,792	3	4	0.973119
30	546	1494	557.41	33,792	3	4	0.97953

Figure 7. (a) Division FPU, ALM, Frequency, Register and Area-Time Products Specs







Latency	0
ALM	24
Register	96
Frequency(MHz)	1333

Table 9. FPU Absolute Value Performance Specs

4.1.5 FPU Absolute Value Unit

Absolute value FPU has only one option. In Table 9, area-frequency values are given.

4.1.6 FPU Square Root Unit

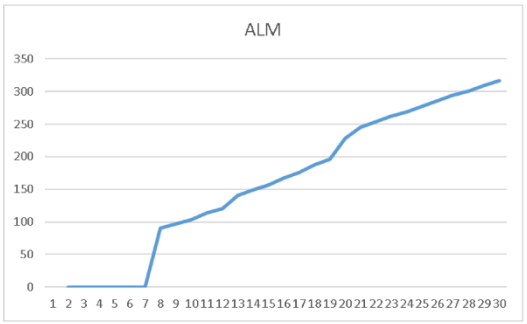
Embedded FPUs cannot be used in square root operations. In addition to ALM and registers, Block RAM and Fixed point processing units are also used. Area performance results are given in Table 10. The line marked with green indicates the highest efficiency FPU.

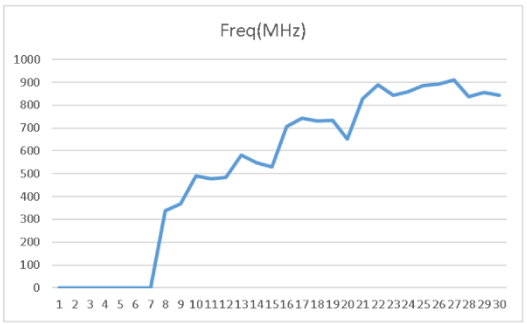
Figure 8 shows the performance effect graphs of the Square Root FPU according to the latency.

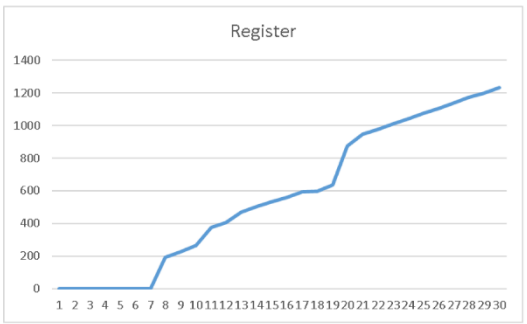
Table 10. Square Root FPU Area - Frequency Specs

Latency	ALM	Register	Freq(MHz)	Block Memory Bits	RAM Blocks	Fixed DSP Blocks	Area-Time Product
1		-	-	-	-	-	-
2	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-
8	91	194	339.33	15,872	3	2	0.268176
9	97	228	368.87	15,872	3	2	0.262965
10	103	266	490.68	15,872	3	2	0.209913
11	114	377	476.87	15,872	3	2	0.239059
12	120	405	484.26	15,872	3	2	0.247801
13	141	469	582.75	15,872	3	2	0.241956
14	149	502	547.35	15,872	3	2	0.272221
15	157	534	530.22	15,872	3	2	0.296104
16	167	557	707.21	15,872	3	2	0.236139
17	176	591	742.39	15,872	3	2	0.237072
18	188	596	731.53	15,872	3	2	0.256996
19	196	636	734.21	15,872	3	2	0.266954
20	228	876	653.17	15,872	3	2	0.349067
21	245	946	829.88	15,872	3	2	0.295223
22	254	977	889.68	15,872	3	2	0.285496
23	262	1010	843.88	15,872	3	2	0.310471
24	269	1042	857.63	15,872	3	2	0.313655
25	277	1074	886.52	15,872	3	2	0.312458
26	286	1106	891.27	15,872	3	2	0.32089
27	294	1140	909.92	15,872	3	2	0.323105
28	301	1171	837.52	15,872	3	2	0.359394
29	309	1200	856.16	15,872	3	2	0.360914
30	317	1234	844.59	15,872	3	2	0.37533

Figure 8. Square Root FPU, ALM, Frequency, Register and Area-Time Products Specs







4.1.7 FPU Comparator Unit

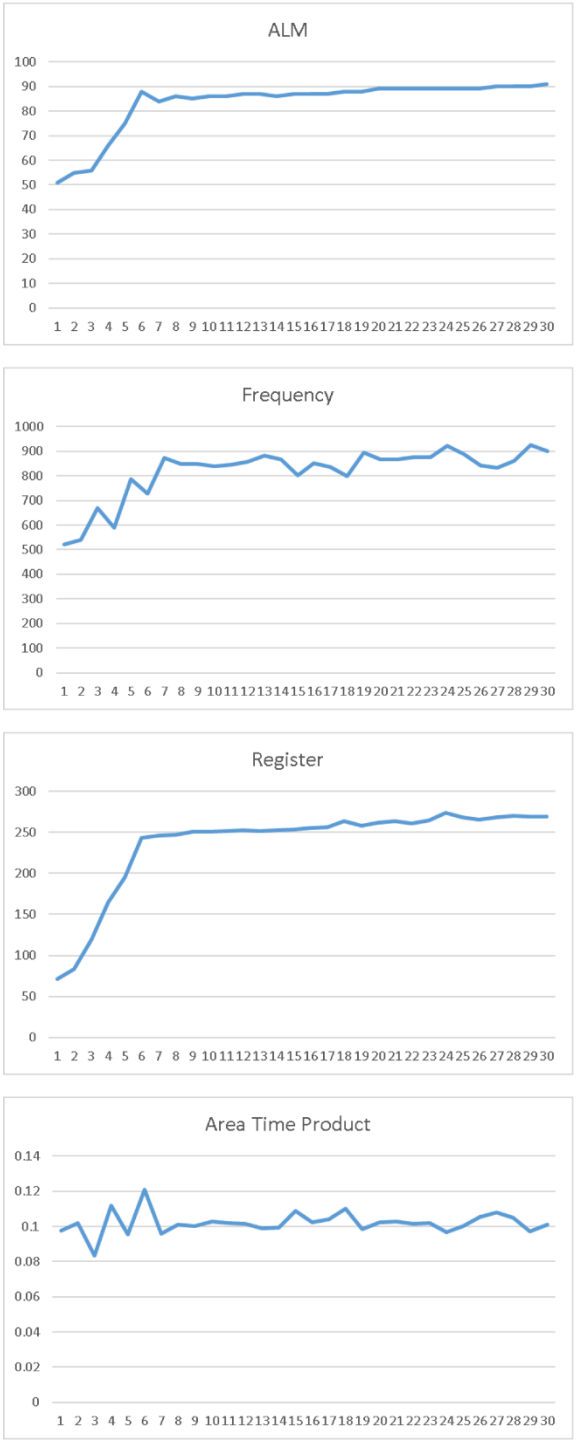
Area-performance results are given in Table 11.

The line marked with green indicates the highest efficiency FPU. Figure 9 shows the performance effect graphs of the Comparator FPU according to the delay.

Table 11. Comparator FPU Area - Frequency Specs

Latency	ALM	Register	Frequency(MHz)	Area-Time Product
1	51	71	521.65	0.097766702
2	55	83	539.96	0.101859397
3	56	119	670.69	0.083496101
4	66	165	590.67	0.111737518
5	75	195	786.16	0.095400427
6	88	243	727.27	0.121000454
7	84	246	874.13	0.096095546
8	86	247	849.62	0.101221723
9	85	251	848.9	0.100129579
10	86	251	837.52	0.102684115
11	86	252	844.59	0.101824554
12	87	253	856.9	0.101528766
13	87	252	881.06	0.098744694
14	86	253	867.3	0.099158307
15	87	254	800.64	0.10866307
16	87	255	850.34	0.102312016
17	87	256	836.12	0.10405205
18	88	264	799.36	0.11008807
19	88	258	895.26	0.098295467
20	89	262	868.06	0.102527475
21	89	264	865.8	0.102795103
22	89	261	877.19	0.101460345
23	89	265	874.89	0.101727074
24	89	274	921.66	0.096564894
25	89	268	887.31	0.100303163
26	89	266	843.26	0.105542774
27	90	268	832.64	0.108089931
28	90	270	859.11	0.104759577
29	90	269	925.07	0.097289935
30	91	269	900.09	0.101101001

Figure 9. Comparator FPU, ALM, Frequency, Register and Area-Time Products Specs



4.1.8 FPU-Fixed Converter Units

Some of the designs require a transformation process for data exchange between units that operate in floating point and some of them with fixed point operation. In this context, the examination of conversion IPs from float to fixed pointe and vice versa, offered by Altera, are given in sub-headings.

4.1.8.1 Float – Fixed Conversion

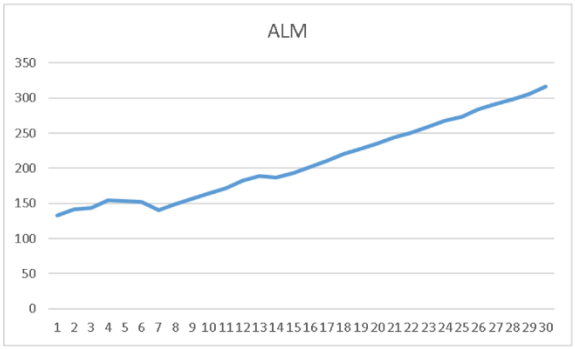
Area performance results are given in Table 12. The line marked with green indicates the highest efficiency IP.

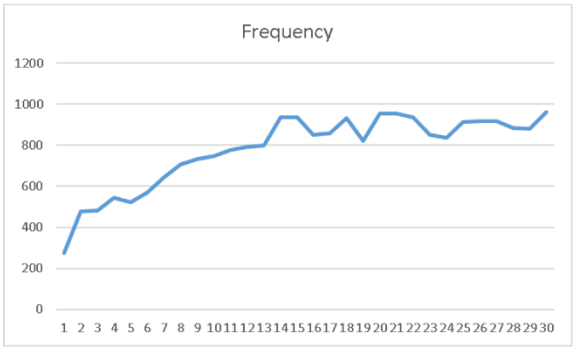
Figure 10 shows the performance effect graphs of the Float-Fixed Converter IP according to the delay.

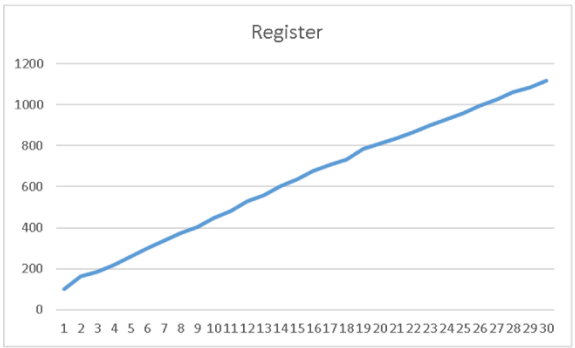
Table 12. LUT Based Converter Unit Specs

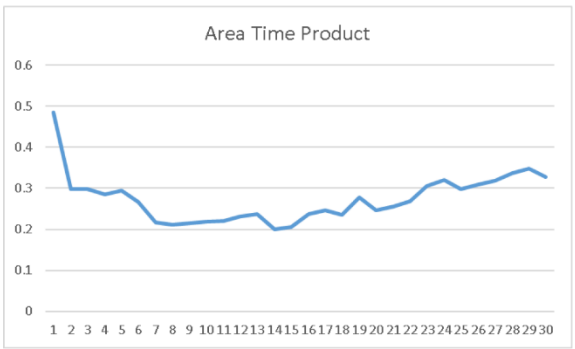
Latency	ALM	Register	Frequency (MHz)	Area Time Product
1	133	101	274.05	0.485313
2	142	165	477.1	0.297632
3	144	186	483.33	0.297933
4	155	220	542.89	0.285509
5	153	259	521.1	0.29361
6	152	299	568.83	0.267215
7	140	337	643.5	0.21756
8	149	376	707.71	0.210538
9	157	405	732.6	0.214305
10	164	449	749.63	0.218775
11	172	483	778.21	0.22102
12	183	531	793.02	0.230763
13	189	558	798.72	0.236629
14	187	602	938.09	0.199341
15	193	636	938.09	0.205737
16	202	676	850.34	0.237552
17	211	707	858.37	0.245815
18	220	733	931.97	0.236059
19	228	783	821.69	0.277477
20	235	809	955.11	0.246045
21	244	836	954.2	0.255712
22	251	867	936.33	0.268068
23	259	898	849.62	0.304842
24	268	929	836.82	0.32026
25	273	960	915.75	0.298116
26	284	995	917.43	0.30956
27	292	1024	918.27	0.317989
28	298	1061	884.96	0.336738
29	306	1084	880.28	0.347617
30	316	1116	962.46	0.328325

Figure 10. Float-Fixed FPU, ALM, Frequency, Register and Area-Time Products Specs









4.1.8.2 Fixed Float Converter Unit

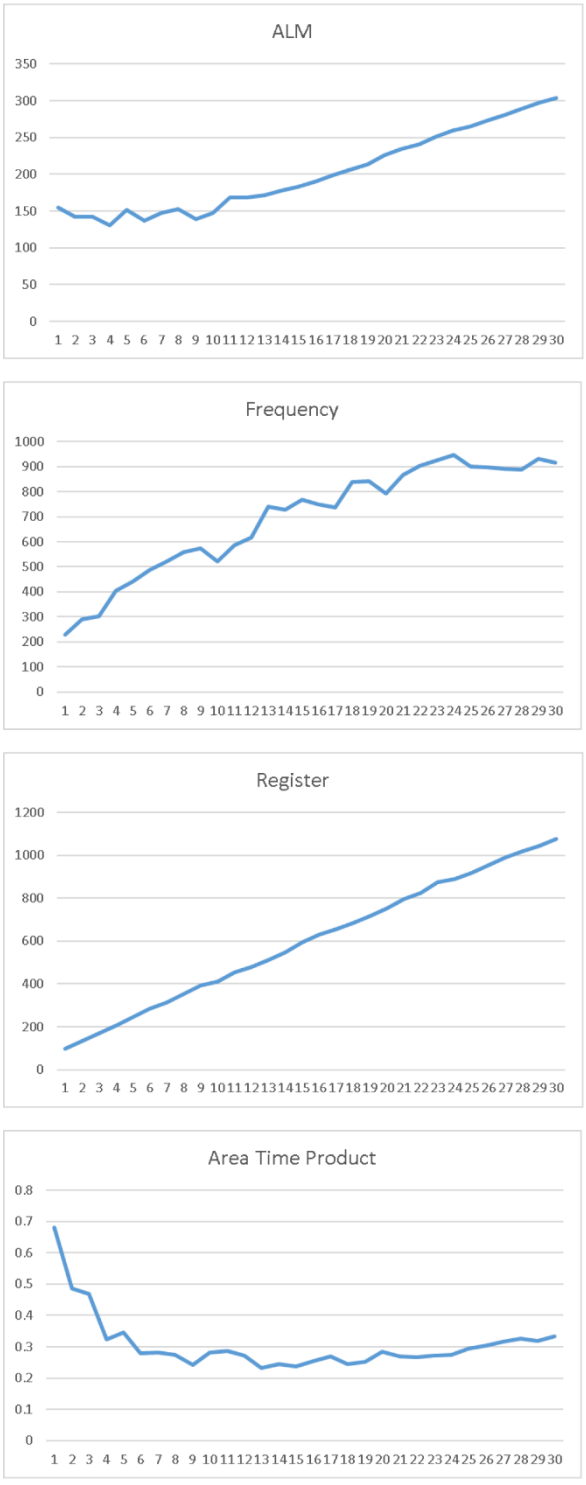
Area performance results are given in Table 13. The line marked with green indicates the highest efficiency IP.

Figure 11 shows the performance effect graphs of the Fixed-Float Converter IP according to the latency.

Table 13. LUT Based Fixed Float Converter Unit Specs

Latency	ALM	Register	Frequency (MHz)	Area Time Product
1	155	96	227.53	0.681229
2	142	135	291.8	0.486635
3	142	169	302.21	0.469872
4	131	205	405.35	0.323178
5	152	244	440.92	0.344734
6	137	286	488.76	0.280301
7	147	314	522.19	0.281507
8	153	353	557.72	0.274331
9	139	394	574.71	0.241861
10	147	412	520.29	0.282535
11	168	454	584.8	0.287278
12	168	480	618.05	0.271823
13	171	510	740.19	0.231022
14	178	547	727.8	0.244573
15	183	595	769.23	0.2379
16	190	631	749.06	0.253651
17	199	657	738.55	0.269447
18	206	683	839.63	0.245346
19	213	718	841.75	0.253044
20	226	754	793.02	0.284987
21	234	795	866.55	0.270036
22	241	825	904.98	0.266304
23	251	875	924.21	0.271583
24	260	889	946.07	0.274821
25	265	919	899.28	0.29468
26	273	953	897.67	0.304121
27	281	990	892.06	0.315001
28	289	1020	887.31	0.325704
29	297	1046	931.1	0.318978
30	304	1078	914.91	0.332273

Figure 11. Fixed-Float FPU, ALM, Frequency, Register and Area-Time Products Specs



Latency	1
ALM	36
Register	128
Frequency(MHz)	610.5

Table 14. Fixed Point Adder Area – Frequency Specs

4.2 Fixed-Point Unit Analysis

Fixed point arithmetic calculation units are addition and multiplication operations. Details of the relevant units are given in the subtitles.

4.2.1 Fixed-Point Adder Unit

There is no IP provided by Intel for fixed point addition. When synthesizing hardware that adds 2 numbers with 32 bits, the results are given in Table 14.

4.2.2 Fixed-Point Multiply Unit

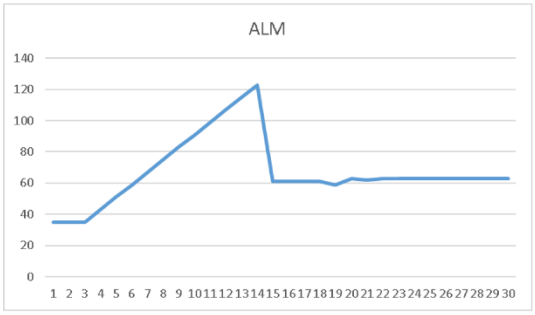
There is an IP provided by Intel for fixed point multiplication. It is possible to synthesize using fixed point DSP or not.

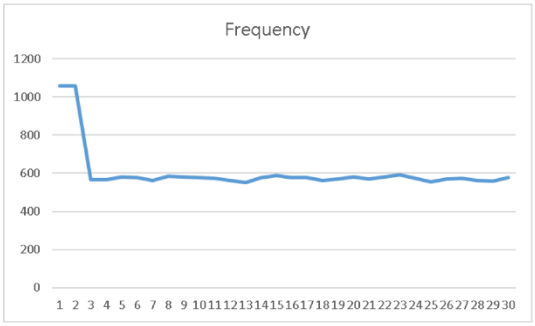
When the synthesis is done using Fixed Point DSPs, the results in Table 15 are obtained. Figure 12 shows the performance effect graphs of Fixed Point Multiplication (with DSP) IP according to the delay.

Table 15. Fixed Point DSP Multiplication Area – Frequency Values

Latency	ALM	Register	Frequency	Fixed DSP Block	Area Time Multiply
1	35	79	1057.08	3	0.03311008
2	35	79	1057.08	3	0.03311008
3	35	111	567.54	3	0.06166966
4	43	143	564.97	3	0.07611024
5	51	175	583.09	3	0.08746506
6	59	207	577.7	3	0.10212913
7	67	239	562.11	3	0.11919375
8	75	271	583.77	3	0.12847526
9	83	303	583.09	3	0.14234509
10	91	335	577.7	3	0.1575212
11	99	367	572.41	3	0.17295295
12	107	399	564.33	3	0.18960537
13	115	431	553.1	3	0.207919
14	123	463	579.37	3	0.21229957
15	61	131	589.62	3	0.10345646
16	61	128	579.37	3	0.10528678
17	61	128	578.03	3	0.10553085
18	61	128	564.65	3	0.10803152
19	59	129	570.45	3	0.10342712
20	63	136	583.09	3	0.10804507
21	62	134	570.78	3	0.10862329
22	63	134	579.71	3	0.10867503
23	63	132	592.42	3	0.10634347
24	63	134	574.05	3	0.10974654
25	63	132	555.25	3	0.1134624
26	63	136	569.8	3	0.11056511
27	63	134	575.37	3	0.10949476
28	63	133	564.65	3	0.11157354
29	63	131	558.04	3	0.11289513
30	63	134	577.7	3	0.10905314

Figure 12. Fixed Point Multiply DSP, ALM, Frequency, Register and Area-Time Products Specs





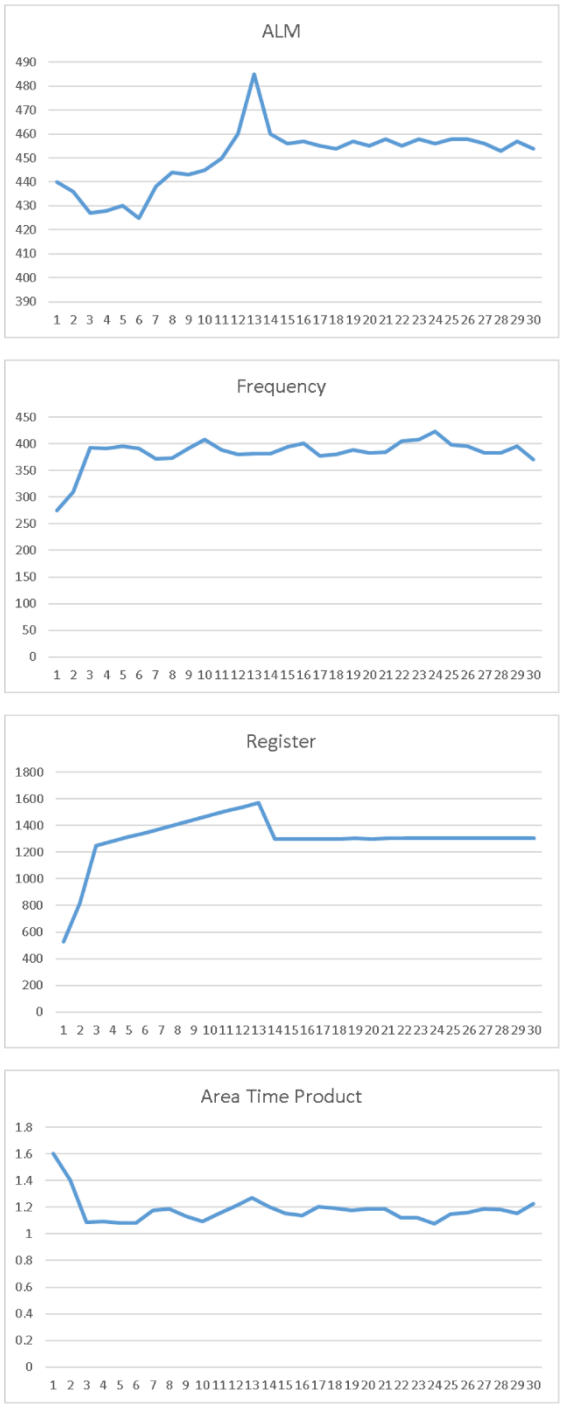
When the fixed point multiplication operation is synthesized without using DSP, the values in Table 16 are taken.

Figure 13 shows the performance effect graphs of Fixed Point Multiplication (without DSP) IP according to latency.

Table 16. Fixed Point Multiply without DSP Area Frequency Specs

Latency	ALM	Register	Frequency	Fixed DSP BLOCKS	Area Time Product
1	440	528	274.35	0	1.603791
2	436	819	310.08	0	1.406089
3	427	1250	393.24	0	1.085851
4	428	1284	391.39	0	1.093538
5	430	1314	396.2	0	1.08531
6	425	1346	392	0	1.084184
7	438	1379	372.3	0	1.176471
8	444	1410	373.97	0	1.187261
9	443	1443	391.7	0	1.130968
10	445	1475	407.66	0	1.091596
11	450	1508	388.95	0	1.156961
12	460	1538	380.08	0	1.210272
13	485	1571	381.24	0	1.272165
14	460	1300	382.12	0	1.20381
15	456	1301	394.01	0	1.157331
16	457	1301	400.48	0	1.141131
17	455	1302	377.07	0	1.206673
18	454	1299	380.95	0	1.191757
19	457	1304	388.5	0	1.176319
20	455	1302	383.44	0	1.186626
21	458	1304	385.06	0	1.189425
22	455	1306	405.52	0	1.122016
23	458	1306	408	0	1.122549
24	456	1304	423.01	0	1.077989
25	458	1304	399.04	0	1.147755
26	458	1307	395.41	0	1.158291
27	456	1303	383.14	0	1.190165
28	453	1303	382.7	0	1.183695
29	457	1306	395.88	0	1.15439
30	454	1303	370.78	0	1.224446

Figure 13. Fixed Point Multiply without DSP, ALM, Frequency, Register and Area-Time Products Specs



5. RESULTS

In this study, an FPU selection methodology is given for FPGA designs which contains floating point operations for the cases where the target is the highest performance. In the FPU selection process, the parameters of the IPs provided by the FPGA manufacturer are determined, while satisfying the minimum area and providing the maximum frequency.

FPUs with the given methodology; the optimum FPU selection process, which can take days or even months, can be reduced to seconds. These techniques have been tested on Arria 10 FPGA, which is a new generation FPGA family of Intel, which is given as an use case. With the backpropagation artificial neural network technique used, the detection rate of the optimum FPU was measured as 93%.

This methodology can also be applied in other FPGA families. It is especially useful for revealing the characteristics of FPGAs that have been newly released, not much work has been done on it and produced with new production technologies.

REFERENCES

Levent, V. E., & Guzel A. E., & Tosun, M., & Buyukmihci, M., & Aydin, F., & Goren, S., & Erbas, C., & Akgun, T., & Ugurdag, H. F. (2018). Tools and Techniques for Implementation of Real-time Video Processing Algorithms, *Springer Journal of Signal Processing Systems (JSPS),* 8(1), 93-113. https://doi.org/10.1007/s11265-018-1402-7

Levent V. E. (2020). Efficient Selection of Floating-Point Units for Maximize a FPGA Based System Throughput, 3. International Conference on Life and Engineering Sciences (ICOLES), İstanbul, Turkey

Uğurdağ H. F. (2013). Experiences On The Road From EDA Developer To Designer To Educator, in Proceedings of the East-West Design & Test Symposium (EWDTS), Rostov, Russia

Intel Arria 10 FPGA Document (https://www.intel.com.tr/content/www/tr/tr/products/programmable/fpga/arria-10.html), Erişim Tarihi: 1.02.2021

P., Schumacher, & P., Jha (2018). Fast and accurate resource estimation of RTL-based designs targeting FPGAs, in Proceedings of the International Conference on Field Programmable Logic and Applications (FPL), Dublin

P., Bjureus, M., Millberg, & A., Jantsch (2002). FPGA resource and timing estimation from matlab execution traces, in Proceedings of the International Symposium on Hardware/Software Codesign (CODES), USA

V., Degalahal, & T., Tuan (2005), Methodology for high level estimation of FPGA power consumption, in Proceedings of the Asia and South Pacic Design Automation Conference (ASP-DAC), China

P., A., Milder, M., Ahmad, J., C., Hoe, & M., Puschel (2006), Fast and accurate resource estimation of automatically generated custom DFT IP cores, in Proceedings of the International Symposium on Field Programmable Gate Arrays (FPGA), USA

C., Shi, J., Hwang, S., McMillan, A., Root, & V., Singh (2004), A system level resource estimation tool for FPGAs, in Proceedings of International Conference on Field Programmable Logic and Applications (FPL), Belgium