



T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ BÖLÜMÜ

EHM4830 PROGRAMLANABİLİR LOJİK DEVRE TASARIMI
PROJE

FLOATING POINT HESAPLAYICI (ÇARPMA– TOPLAMA)

Dersi Veren Öğretim Üyesi
Prof.Dr. Burcu ERKMEN

Ödevi Yapan Öğrenci
Murat DELİBALTA
18014011

Ödev Teslim Tarihi
29.05.2022

İÇİNDEKİLER

İÇİNDEKİLER	1
KISALTMA LİSTESİ.....	2
ŞEKİL LİSTESİ.....	3
ÇİZELGE LİSTESİ	4
ÖZET	5
BÖLÜM 1	6
1. GİRİŞ	6
1.1 Floating Point	6
BÖLÜM 2	9
2. GELİŞME	9
2.1 Floating Point Toplama İşlemi	9
2.2 Floating Point Çarpma İşlemi	13
2.3 Floating Point Toplama İşlemi Simülasyon Sonucu	16
2.4 Floating Point Çarpma İşlemi Simülasyon Sonucu	16
2.5 Floating Point Top Modül Simülasyon Sonucu	16
BÖLÜM 3	17
3. SONUÇ.....	17
3.1 VHDL Modüllerinin FPGA Kiti Test Sonuçları	18
3.1.1 Toplama Devresi	18
3.1.2 Çarpma Devresi.....	21
KAYNAKLAR	23

e	exponent biti
FPGA	Field Programmable Gate Array
IEEE	Institute of Electrical and Electronics Engineers
m	mantissa biti
NaN	Not a Number
s	işaret biti
VHDL	Very High Speed Integrated Circuit Hardware Description Language

	Sayfa
Şekil 1.1 Floating Point Gösterim.....	7
Şekil 1.2 0.15625 Ondalıklı Sayısının Floating Point Gösterimi [3]	7
Şekil 2.1 Floating Point Toplama İşlemi Simülasyon Sonucu	16
Şekil 2.2 Floating Point Toplama İşlemi Simülasyon Sonucu	16
Şekil 2.3 Floating Point Top Modül Simülasyon Sonucu	16
Şekil 3.1 Örnek 1– $1,5 + 0,5 = 2$	18
Şekil 3.2 Örnek 2– $1.5 - 0.5 = 1$	18
Şekil 3.3 Örnek 3- $1.375 - 0.875 = 0.5$	19
Şekil 3.4 Örnek 4- $1.375 + 0.875 = 2.25$	19
Şekil 3.5 Örnek 5- $0.375 - 1.75 = -1.375$	20
Şekil 3.6 Örnek 6- $0.375 + 1.75 = 2.125$	20
Şekil 3.7 Örnek 1- $-1.5 \times 2.5 = -3.75$	21
Şekil 3.8 Örnek 2- $2.5 \times 0.25 = 1$	21
Şekil 3.9 Örnek 3- $1.375 \times -0.875 = -1.203125$	22
Şekil 3.10 Örnek 4- $0.375 \times -1.75 = -0.65625$	22

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 2.1 1.5 Sayısının Floating Point Gösterimi	9
Çizelge 2.2 0.5 Sayısının Floating Point Gösterimi	10
Çizelge 2.3 2 Sayısının Floating Point Gösterimi	10
Çizelge 2.4 1 Sayısının Floating Point Gösterimi	10
Çizelge 2.5 1.375 Sayısının Floating Point Gösterimi	11
Çizelge 2.6 0.875 Sayısının Floating Point Gösterimi	11
Çizelge 2.7 2.25 Sayısının Floating Point Gösterimi	11
Çizelge 2.8 0.375 Sayısının Floating Point Gösterimi	12
Çizelge 2.9 -1.75 Sayısının Floating Point Gösterimi	12
Çizelge 2.10 -1.375 Sayısının Floating Point Gösterimi	12
Çizelge 2.11 2.125 Sayısının Floating Point Gösterimi	12
Çizelge 2.12 -1.5 Sayısının Floating Point Gösterimi	13
Çizelge 2.13 2.5 Sayısının Floating Point Gösterimi	14
Çizelge 2.14 -3.75 Sayısının Floating Point Gösterimi	14
Çizelge 2.15 0.25 Sayısının Floating Point Gösterimi	14
Çizelge 2.16 -1.203125 Sayısının Floating Point Gösterimi	15
Çizelge 2.17 -0.65625 Sayısının Floating Point Gösterimi	15

FLOATING POINT HESAPLAYICI(ÇARPMA TOPLAMA)

Floating Point aritmetiği, modern bilgisayarlarda sayısal hesaplamalar yapmak için gerçek sayı aritmetiğine yaklaşmanın açık ara en çok kullanılan yoludur. Uzunca bir süre her bilgisayar farklı bir aritmetiğe sahipti. Bunlar tabanlar, anlamlılık ve üslerin boyutları, biçimleri vb. olarak sıralanabilir. Geçmişte, her şirket kendine uygun olan modeli uygulamış ve tek ve evrensel bir standart tanımlayan IEEE 754 standardı ortaya çıkana kadar farklı ekipmanlar arasında taşınabilirliği engellemiştir.

Floating Point hakkında detaylı bilgi verilmiştir. Floating Pointlerle ilgili toplama ve çarpma işlemleri yapılmıştır ve örneklendirilmiştir. Bu örnekler sonrasında Vivado programında VHDL kodları yazılmıştır. Yazılan VHDL kodlarının testbench dosyaları oluşturulmuştur. Oluşturulan testbench dosyalarından simülasyon çıktıları alınmıştır.

Sonrasında Belirtilen Floating Point örnekleri hem çarpma hem de toplama işlemlerini kapsayacak şekilde Xilinx Artix-7 (NEXYS A7) FPGA kitinde test edilmiştir. Bu FPGA kitinde toplamda 16 tane switch olduğundan ötürü 8-bitlik floating point çarpıcı ve toplayıcı devre tasarlanmıştır. Projede 8 bitlik floating pointi ifade etmek için 1 işaret biti, 3 exponent biti, ve 4 bit mantissa kullanılmıştır.

Anahtar Kelimeler: Floating Point, FPGA, Floating Point Toplama, Floating Point Çarpma

1.1 Floating Point

Floating Point aritmetiği, modern bilgisayarlarda sayısal hesaplamalar yapmak için gerçek sayı aritmetiğine yaklaşmanın açık ara en çok kullanılan yoludur. Uzunca bir süre her bilgisayar farklı bir aritmetiğe sahipti. Bunlar tabanlar, anlamlılık ve üslerin boyutları, biçimleri vb. olarak sıralanabilir. Geçmişte, her şirket kendine uygun olan modeli uygulamış ve tek ve evrensel bir standart tanımlayan IEEE 754 standardı ortaya çıkana kadar farklı ekipmanlar arasında taşınabilirliği engellemiştir [1].

Floating Point'in, Fixed Point'ten farkı ondalık işaretinin sabit olmamasıdır. Fixed point sayılar binary formda xx,xxx şeklinde saklanıp ondalık tabanda,

$$\sum_{n=0}^p m_n \times 2^{k-n}, \quad p, k \in \mathbb{Z}^+$$

şeklinde ifade edilir. Floating pointte ise eksponansiyel olarak $(-1)^s \times m \times 2^e$ şeklinde ifade edilir. Buradaki harfler;

s: significant (işaret biti)

e: exponent (üs bitleri)

m: mantissa

olarak adlandırılır. Floating sayılar binary formda ise şu şekilde saklanır:

$$\underbrace{s}_{\text{significand}} \underbrace{e_{k-1} e_{k-2} \dots e_0}_{\text{exponent}} \underbrace{m_{p-1} m_{p-2} \dots m_0}_{\text{mantissa}}$$

Ayrıca exponent ve mantissa bitlerinin sayısı değiştirilerek hassasiyet ve sınır ayarlanabilmektedir. Floating point sayılar onluk tabanda,

$$\left(1 + \sum_{n=1}^{k-1} m_n \times 2^{-n} \right) \times 2^{(e-\text{bias})}$$

şeklinde ifade edilir. Burada belirtilen bias değeri, k exponent biti için 2^{k-1} olmaktadır.

IEEE754 standartındaki floating point formatları:

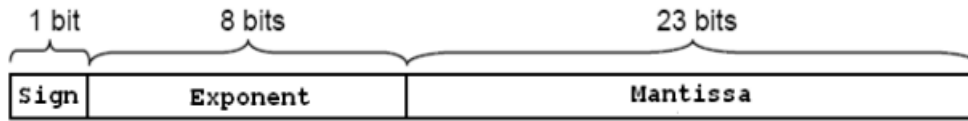
Half Precision: 5 Exponent biti, 10 Mantissa biti. Onluk tabanda ≈ 3.3 basamak

Single Precision: 8 Exponent biti, 23 Mantissa biti. Onluk tabanda ≈ 7.2 basamak.

Double Precision: 11 Exponent biti, 52 Mantissa biti. Onluk tabanda ≈ 15.9 basamak.

Quad Precision: 15 Exponent biti, 112 Mantissa biti. Onluk tabanda ≈ 34 basamak.

olarak 4 farklı şekilde ifade edilebilmektedir.



Şekil 1.1 Floating Point Gösterim

Exponent ile ilgili 3 tane özel durum vardır:

Denormalize Sayı: Exponent'in 0 olması durumudur. Bu durumda sayı onluk tabanda,

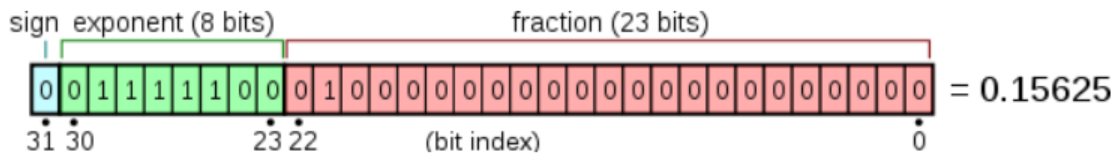
$$\left(\sum_{n=1}^{k-1} m_n \times 2^{-n} \right) \times 2^{(e-bias)}$$

şeklinde ifade edilir.

Sonsuz: Exponent'in bütün bitlerinin '1', mantissa'nın ise '0' olması durumudur.

NaN (Not a Number): Exponentin bütün bitlerinin '1' ve mantissa'nın ise '0' 'dan farklı olması durumudur [2].

Single Precision bir örnek inceleyecek olursak;



Şekil 1.2 0.15625 Ondalık Sayısının Floating Point Gösterimi [3]

Şekil 1.2'de 0.15625 sayısının floating pointte gösterimi verilmiştir. Fraction ile mantissa aynı kavramları ifade etmek için kullanılmaktadır. 32 bitlik bir floating pointi 1 bit işaret,

8 bit exponent ve 23 bit mantissa biti bulunmaktadır. Bu dönüşümü aşama aşama inceleyecek olursak 0.15625 sayısı;

$$0.15625 = 0 \times 0.5 + 0 \times 0.25 + 1 \times 0.125 + 0 \times 0.0625 + 1 \times 0.03125$$

olarak ifade edilmektedir. Binary kodladığımız zaman ise 0.00101'e denk gelmektedir. Sayıyı saya doğru 3 kere kaydığımızda 1.01×2^{-3} elde edilmektedir. Yani mantissa (fraction) değerimiz 01 olur. Exponent 8 bit olduğundan ötürü $2^{8-1} - 1$ 'den 127 bias değeri eklenir. $-3 + 127 = 124$ olur. Bu değer de $(01111100)_2$ binary değerine karşılık gelmektedir. Son olarak işaretin kontrol edilmesi gerekmektedir. Sayımız pozitif olduğundan ötürü sign bitimiz '0' olur. Üçünün birleşiminde ise $(1\ 01111100\ 010000000000000000000000)_2$ binary değeri elde edilmiş olur.

Bu projede Floating Point değerleri hesaplanarak hem çarpma hem de toplama işlemleri Xilinx Artix-7 (NEXYS A7) FPGA kitinde test edilmiştir. Bu FPGA kitinde toplamda 16 tane switch olduğundan ötürü 8-bitlik floating point çarpıcı ve toplayıcı devre tasarlanmıştır. Bu şekilde yazılan VHDL modülü sadece testbench üzerinde kalmayıp aynı zamanda BOARD üzerinde de gerçekleşmiş olacaktır. Projede 8 bitlik floating pointi ifade etmek için 1 işaret biti, 3 exponent biti, ve 4 bit mantissa kullanılmıştır.

2.1 Floating Point Toplama İşlemi

Floating Point sayılar toplanırken sırasıyla izlenmesi gereken adımlar şunlardır:

1. Toplanacak olan sayılardan küçük olan sayının anlamlı kısmını, üstler eşitleninceye kadar sağa kaydır.
2. Anlamlı kısımları topla.
3. Gerekiyorsa sayıyı olağanlaştır.
4. Anlamlı kısmı, gösterimde belirlenmiş olan bit sayısına yuvarla.
5. Sonuç olağan hale gelene kadar son iki adımı tekrarla [4].

Bu aşamalar tekrarlandığı taktirde sonuç elde edilmektedir. Sırayla örnekler inceleyelim.

Örnek 1.

$$1.5 + 0.5 = 2$$

1.5 sayısını ifade etmek için;

$(1.1)_2$ şeklinde ifade edilir.

Floating Point Gösterimi ise şu şekildedir:

sign	exp_2	exp_1	exp_0	mat_3	mat_2	mat_1	mat_0
0	0	1	1	1	0	0	0

Çizelge 2.1 1.5 Sayısının Floating Point Gösterimi

Binary sayımız + olduğundan ötürü sign '0' değerini alır.

Sayının exponent kısmı 0 olduğundan ötürü $0+3 = 3$ olmuştur.

Virgülden sonraki kısımları ise görülebileceği gibi mat_3'ten itibaren yazılmıştır.

0.5 sayısı ise:

$(0.1)_2$ şeklinde ifade edilir.

Floating Point gösterimi:

sign	exp_2	exp_1	exp_0	mat_3	mat_2	mat_1	mat_0
0	0	1	0	0	0	0	0

Çizelge 2.2 0.5 Sayısının Floating Point Gösterimi

Çizelge 2.2'deki gibi olmaktadır.

2 sayısının binary gösterimi ise:

1.00×2^1 şeklinde olmaktadır. Floating gösterimi ise Çizelge 2.3'teki gibidir.

sign	exp_2	exp_1	exp_0	mat_3	mat_2	mat_1	mat_0
0	1	0	0	0	0	0	0

Çizelge 2.3 2 Sayısının Floating Point Gösterimi

Örnek 2.

$$1.5 - 0.5 = 1$$

Bu işlemin floating point gösterimleri bir önceki örneğe çok yakındır. 1.5 için Çizelge 2.1'deki ile aynı sonuç elde edilecek olup, 0.5 sayısı için Çizelge 2.2'deki tablodan tek fark sign biti '0' yerine '1' olacaktır. Bu bilgiler eşliğinde sonucumuz şu şekilde olur.

$(1.0)_2$

sign	exp_2	exp_1	exp_0	mat_3	mat_2	mat_1	mat_0
0	0	1	1	0	0	0	0

Çizelge 2.4 1 Sayısının Floating Point Gösterimi

Örnek 3.

$$1.375 + 0.875 = 2.25$$

1.375 sayısı;

$(1.011)_2$ şeklinde ifade edilir. Floating Point gösterimi ise Çizelge 2.5'teki gibidir.

sign	exp_2	exp_1	exp_0	mat_3	mat_2	mat_1	mat_0
0	0	1	1	0	1	1	0

Çizelge 2.5 1.375 Sayısının Floating Point Gösterimi

0.875 sayısı;

$(0.111)_2 = (1.11)_2 \times 2^{-1}$ şeklindedir. Floating Point gösterimi ise Çizelge 2.6'dadır.

sign	exp_2	exp_1	exp_0	mat_3	mat_2	mat_1	mat_0
0	0	1	0	1	1	0	0

Çizelge 2.6 0.875 Sayısının Floating Point Gösterimi

Sonuç 2.25 olması gerekmektedir.

$$(10.01)_2 = (1.001)_2 \times 2^1$$

2.25 sayısının Floating Point gösterimi ise Çizelge 2.7'deki gibidir.

sign	exp_2	exp_1	exp_0	mat_3	mat_2	mat_1	mat_0
0	1	0	0	0	0	1	0

Çizelge 2.7 2.25 Sayısının Floating Point Gösterimi

Örnek 4.

$$1.375 - 0.875 = 0.5$$

1.375 için gerekli örnekleme Çizelge 2.5'te verilmiştir. -0.875 sayısı için de tek değişmesi gereken Çizelge 2.6'daki işaret bitini '0'dan '1' yapmaktır. Bunun sonucunda 0.5 değeri elde edilmektedir. Bu sonuç da Çizelge 2.2'de bulunmaktadır.

Örnek 5.

$$0.375 - 1.75 = -1.375$$

0.375 sayısı;

$(0.011)_2 = (1.1)_2 \times 2^{-2}$ olur. Floating Point gösterimi ise Çizelge 2.8'deki gibi olur.

sign	exp_2	exp_1	exp_0	mat_3	mat_2	mat_1	mat_0

0	0	0	1	1	0	0	0
---	---	---	---	---	---	---	---

Çizelge 2.8 0.375 Sayısının Floating Point Gösterimi

-1.75 sayısı;

$-(1.11)_2$ olur. Floating Point gösterimi ise:

sign	exp_2	exp_1	exp_0	mat_3	mat_2	mat_1	mat_0
1	0	1	1	1	1	0	0

Çizelge 2.9 -1.75 Sayısının Floating Point Gösterimi

Sonuç -1.375 elde edilmektedir.

$-(1.011)_2$ olur. Floating Point gösterimi ise Çizelge 2.10'daki gibidir.

sign	exp_2	exp_1	exp_0	mat_3	mat_2	mat_1	mat_0
1	0	1	1	0	1	1	0

Çizelge 2.10 -1.375 Sayısının Floating Point Gösterimi

Örnek 6.

$$0.375 + 1.75 = 2.125$$

0.375 değeri Çizelge 2.8'de belirtilmiştir. +1.75 ise Çizelge 2.9'da bulunan çizelgenin sign bitinin '0' olmuş halidir. Çıkış ise şekildeki gibi olmaktadır.

$$2.125$$

$$(10.001)_2 = (1.0001)_2 \times 2^1$$

sign	exp_2	exp_1	exp_0	mat_3	mat_2	mat_1	mat_0
0	1	0	0	0	0	0	1

Çizelge 2.11 2.125 Sayısının Floating Point Gösterimi

Bu 6 örnek aynı zamanda VHDL ile kodlanmıştır ve testbench dosyaları oluşturulmuştur. Ayrıca XILINX ARTIX-/ FPGA geliştirme kitinde gerçekleştirilmiş olup bu gerçeklemeler de bir sonraki bölümde örneklenmiştir.

2.2 Floating Point Çarpma İşlemi

Kayan noktalı sayılarda çarpma işlemi yapılırken;

1. İki sayının üst değerleri toplanarak sonucun üst değeri elde edilir. (Ancak bu üst değerleri saptırılmış olduğundan, sonucun üst değeri iki kez saptırılmış olarak elde edilir. Bu nedenle sonucun üst değerinden saptırma değeri çıkarılarak gerçek üst değeri bulunur)
2. İki sayının anlamlı kısımları çarpılarak sonucun anlamlı kısmı hesaplanır.
3. Sonuç olağanlaştırılır.
4. Bit sayısına göre yuvarlama yapılır [4].

Görselleştirecek olursak;

$$f_3 = f_1 \times f_2$$

$$f_3 = \left[(-1)^{s_1} \times m_1 \times 2^{e_1} \right] \times \left[(-1)^{s_2} \times m_2 \times 2^{e_2} \right]$$

$$f_3 = (-1)^{s_1 \oplus s_2} \times (m_1 \times m_2) \times 2^{e_1 + e_2}$$

Şeklinde ifade edilir. Görüldüğü gibi significant bitlerinin XOR işlemine sokulması, mantissaların çarpılması ve exponentlerin toplanması ile çarpma işlemi gerçekleştirilir.

Örnek 1.

$$-1.5 \times 2.5 = -3.75$$

-1.5 sayısını ifade etmek için;

$-(1.1)_2$ şeklinde ifade edilir.

Floating Point Gösterimi ise şu şekildedir:

sign	exp_2	exp_1	exp_0	mat_3	mat_2	mat_1	mat_0
1	0	1	1	1	0	0	0

Çizelge 2.12 -1.5 Sayısının Floating Point Gösterimi

2.5 sayısını ifade etmek için;

$(10.1)_2 = (1.01)_2 \times 2^1$ şeklindedir. Floating Point gösterimi;

sign	exp_2	exp_1	exp_0	mat_3	mat_2	mat_1	mat_0
0	1	0	0	0	1	0	0

Çizelge 2.13 2.5 Sayısının Floating Point Gösterimi

şeklindedir. Sonuç ise;

-3.75

$(-11,11)_2 = (1.0001)_2 \times 2^1$ şeklindedir. Floating Point gösterimi ise Çizelge 2.14'tedir.

Sign	exp_2	exp_1	exp_0	mat_3	mat_2	mat_1	mat_0
1	1	0	0	1	1	1	0

Çizelge 2.14 -3.75 Sayısının Floating Point Gösterimi

Sayıları ters çevirip aynı işlemi tekrarlamaya gerek bulunmamaktadır. Çünkü işaretler çarpıldığından ötürü zıt işaretlerin çarpımı '1' sonucunu, aynı işaretlerin çarpımı '0' işaret bitini verecektir

Örnek 2.

$$2.5 \times 0.25 = 1$$

2.5 değeri Çizelge 2.13'te verilmiştir.

0.25 için;

$(0,01)_2 = (1.0)_2 \times 2^{-2}$ şeklindedir. Floating Point gösterimi ise Çizelge 2.15'tedir.

Sign	exp_2	exp_1	exp_0	mat_3	mat_2	mat_1	mat_0
0	0	0	1	0	0	0	0

Çizelge 2.15 0.25 Sayısının Floating Point Gösterimi

Çıkış 1 olup, Çizelge 2.4'te mevcuttur.

Örnek 3.

$$1.375 \times -0.875 = -1.203125$$

1.375 için gerekli çizelge ve bilgiler Çizelge 2.5'te mevcuttur. -0.875 için ise Çizelge 2.6'da bulunan çizelgenin sadece sign bitinin değişmesiyle elde edilmektedir.

-1.203125 için;

$-(1,001101)_2$ şeklindedir. Buradaki örnek incelendiğinde diğer örneklere göre farklıdır. Çıkışımız 8 bit ve sadece 4 biti mantissa'dan oluştuğundan ötürü buradaki mantissa bitlerimizin sadece en yüksek 4 biti alınacak olup diğer 2 bit'i alınmayacaktır Floating Point gösterimi ise Çizelge 2.16'tedir.

Sign	exp_2	exp_1	exp_0	mat_3	mat_2	mat_1	mat_0
1	0	1	1	0	0	1	1

Çizelge 2.16 -1.203125 Sayısının Floating Point Gösterimi

Örnek 4.

$$0.375 \times -1.75 = -0.65625$$

0.375 ve -1.75 değerleri sırasıyla Çizelge 2.8 ve Çizelge 2.9'da belirtilmiştir. Bu çizelgelere istinaden çıkış tablosu;

-0.65625

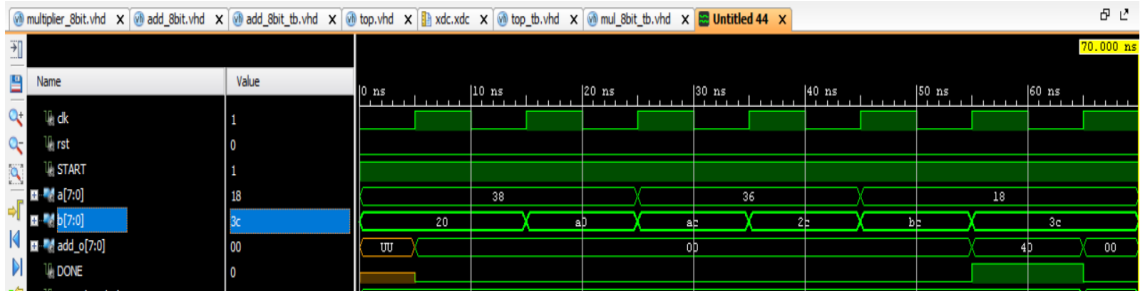
$(0,10101)_2 = (1.0101)_2 \times 2^{-1}$ şeklindedir. Floating Point gösterimi ise Çizelge 2.17'tedir.

Sign	exp_2	exp_1	exp_0	mat_3	mat_2	mat_1	mat_0
1	0	1	0	0	1	0	1

Çizelge 2.17 -0.65625 Sayısının Floating Point Gösterimi

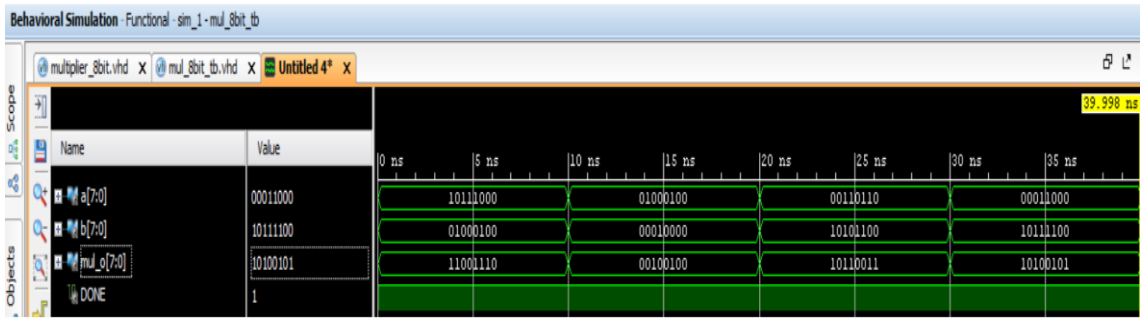
Bu 4 örnek aynı zamanda VHDL ile kodlanmıştır ve testbench dosyaları oluşturulmuştur. Ayrıca XILINX ARTIX-/ FPGA geliştirme kitinde gerçekleştirilmiş olup bu gerçeklemeler de bir sonraki bölümde örneklendirilmiştir.

2.3 Floating Point Toplama İşlemi Simülasyon Sonucu



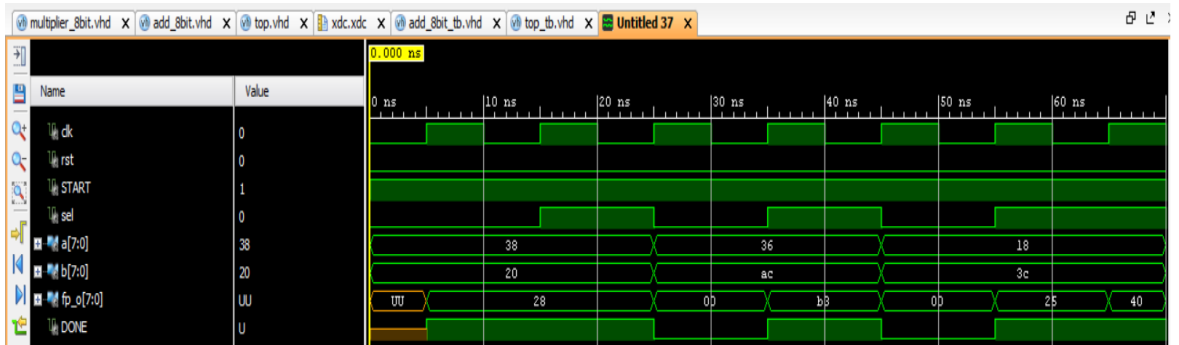
Şekil 2.1 Floating Point Toplama İşlemi Simülasyon Sonucu

2.4 Floating Point Çarpma İşlemi Simülasyon Sonucu



Şekil 2.2 Floating Point Toplama İşlemi Simülasyon Sonucu

2.5 Floating Point Top Modül Simülasyon Sonucu



Şekil 2.3 Floating Point Top Modül Simülasyon Sonucu

SONUÇ

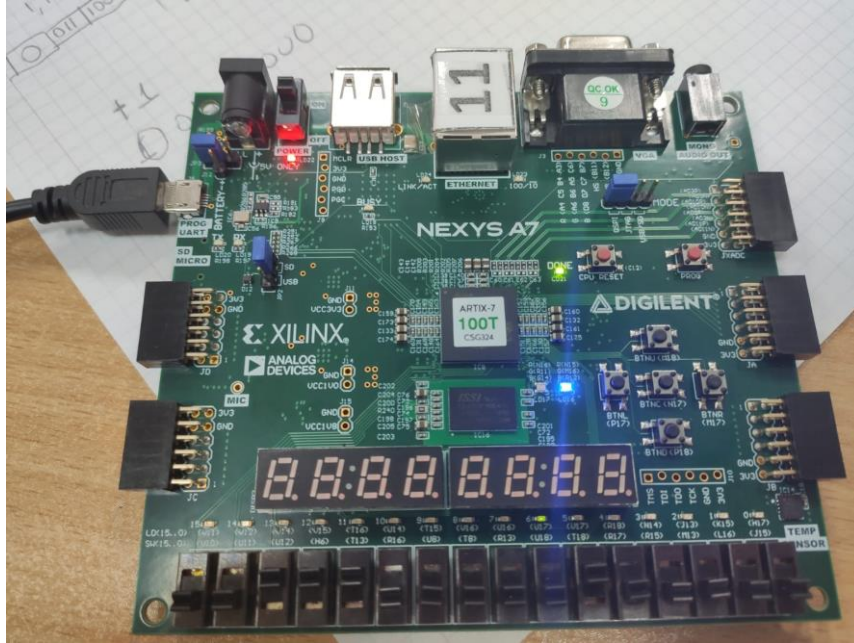
Projede ilk olarak Floating Point kavramı derince araştırılmıştır. Yapılan araştırmalar sonucu elde edilen bilgiler ve üzerine katılan yorumlarla giriş bölümü oluşturulmuştur. Burada ilk olarak Floating Point'in, Fixed Point'ten farkına değinilmiştir. Devamında ise Floating Point'in yapısı detaylıca anlatılmıştır. IEEE754 standardına değinilmiş ve Single Precision ile örnekleme yapılmıştır. Son olarak neden 8-bit devam edileceğine değinilmiş ve sonraki bölüme geçilmiştir.

2. Bölümde ilk olarak Floating Point Toplama işleminden bahsedilmiştir. Devamında ise örnekler yapılarak konunun daha iyi anlaşılması sağlanmıştır. Sonrasında Floating Point Çarpma işlemiyle devam edilip aynı şekilde örneklerle pekiştirilmiştir. Bu aşamadan sonra VHDL modül kısımları gelmektedir. İlk başta sırasıyla Floating Point Toplayıcı VHDL modülü, testbenchi ve simülasyon sonucu verilmiştir. Sonrasında Floating Point Çarpma işlemi VHDL modülü, testbenchi ve simülasyonu verilmiştir. Son olarak bu 2 modül top modülde birleştirilmiş ve bu modülün kodları ve simülasyonuyla projenin bu bölümü sonlanmıştır.

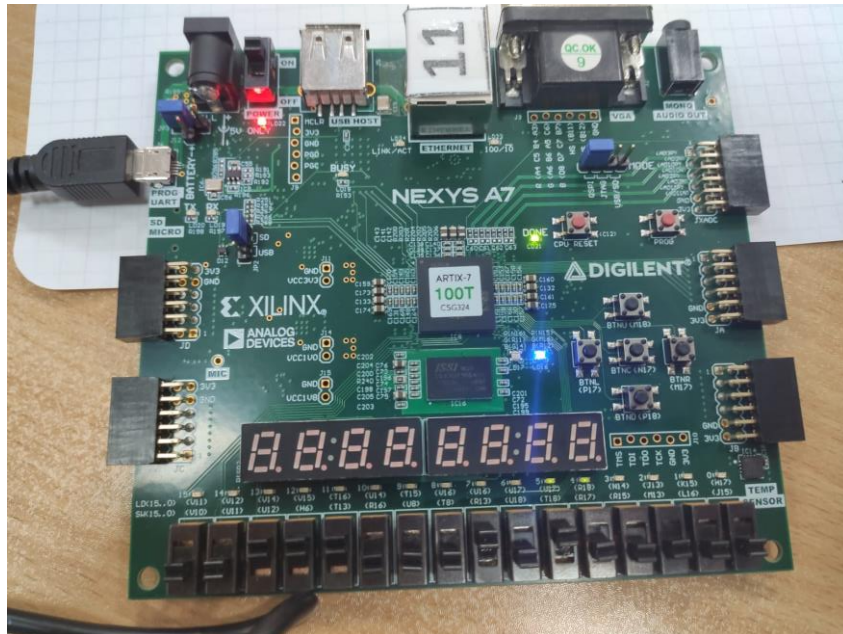
Son Bölümde ise projenin en önemli kısmı olan VHDL modüllerini FPGA kitine gömme işlemi tamamlanmıştır. Bu işlem sonrasında FPGA kartında oluşan girdi ve çıktıların tamamı bu bölümün devamında verilmiştir.

3.1 VHDL Modüllerinin FPGA Kiti Test Sonuçları

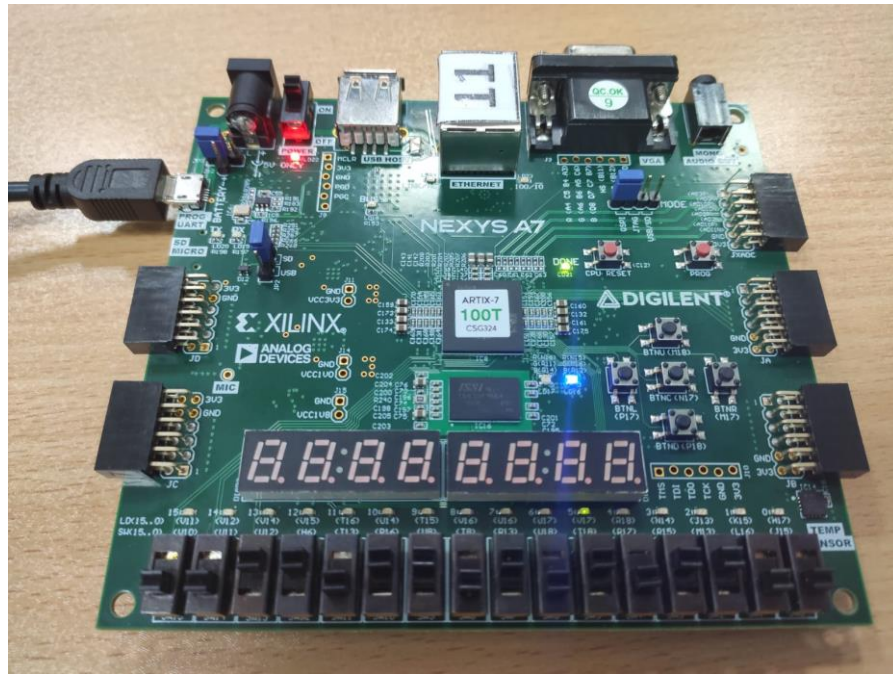
3.1.1 Toplama Devresi



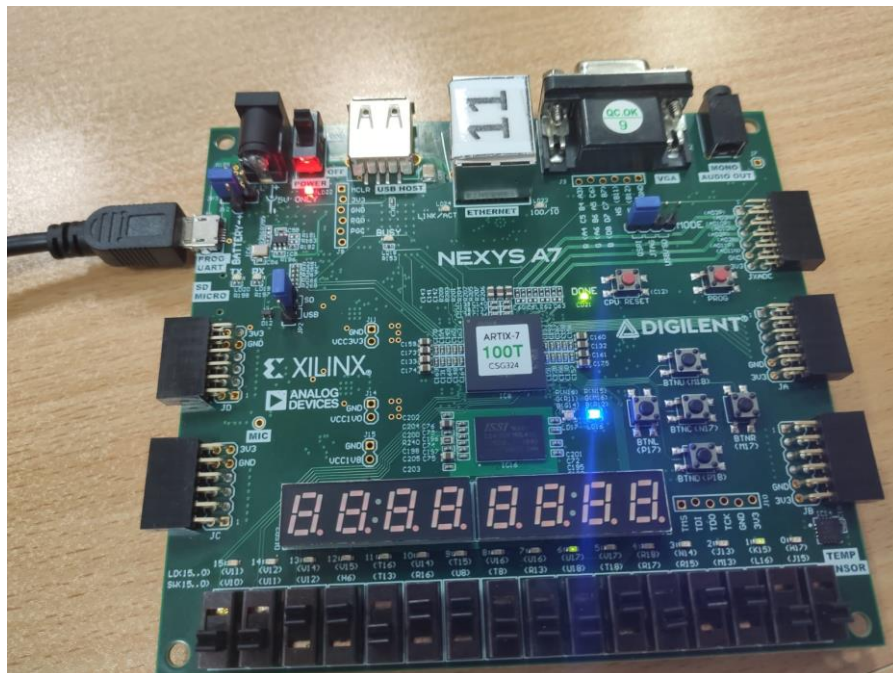
Şekil 3.1 Örnek 1– $1,5 + 0,5 = 2$



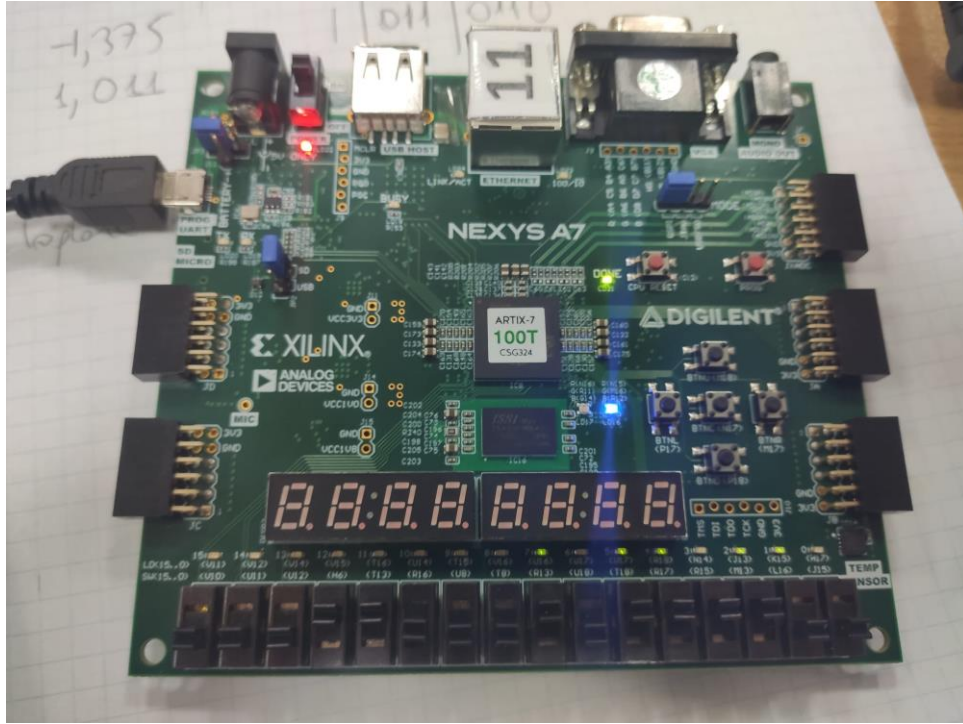
Şekil 3.2 Örnek 2– $1,5 - 0,5 = 1$



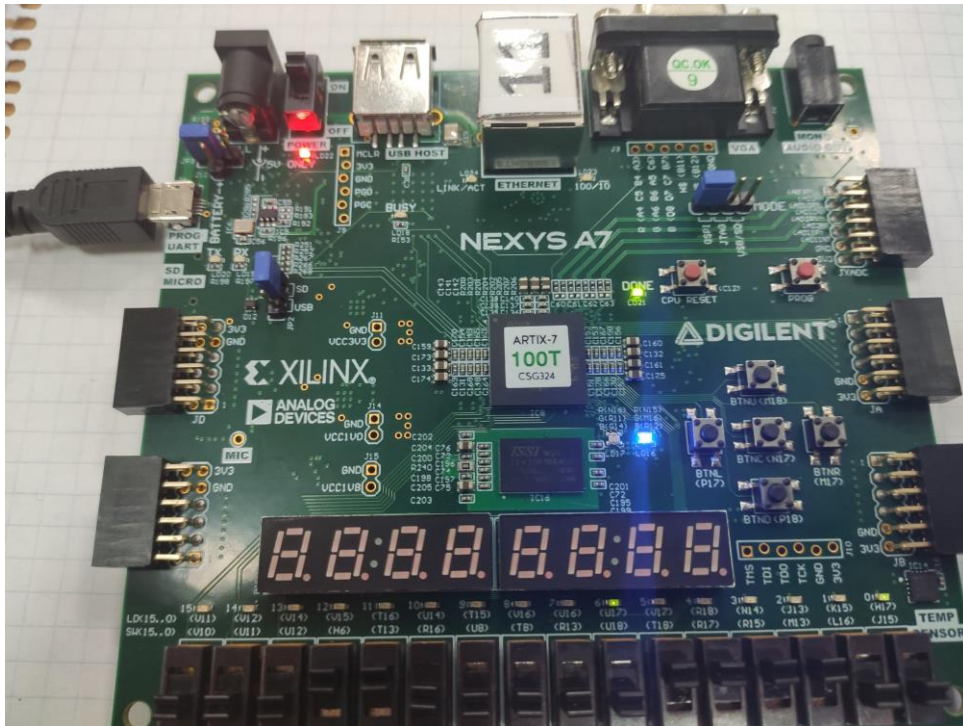
Şekil 3.3 Örnek 3- $1.375 - 0.875 = 0.5$



Şekil 3.4 Örnek 4- $1.375 + 0.875 = 2.25$

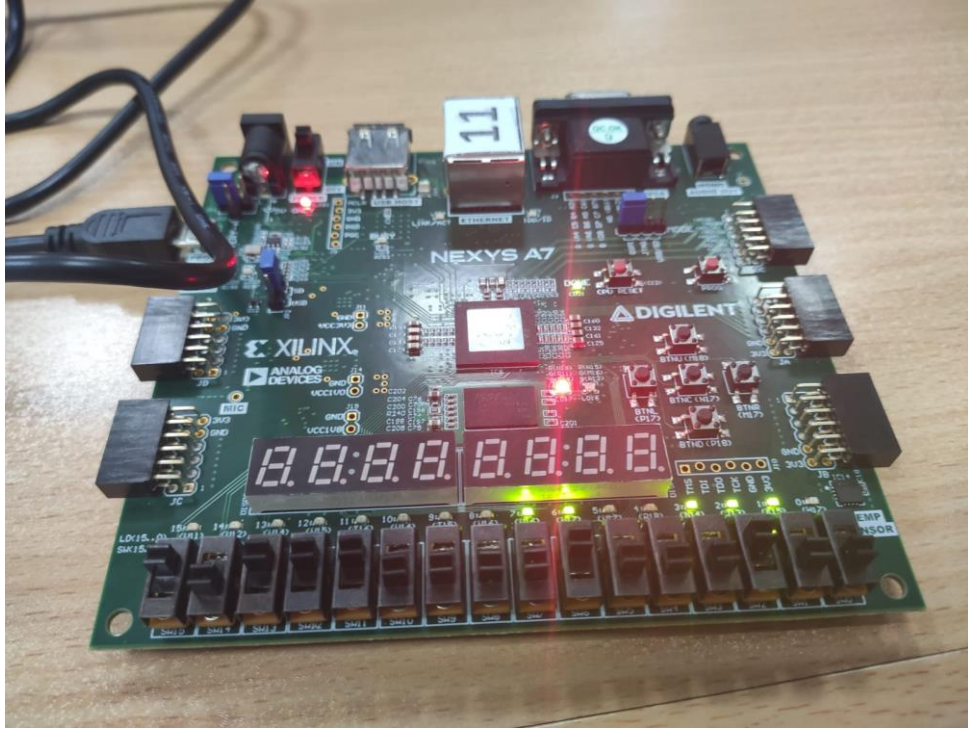


Şekil 3.5 Örnek 5- $0.375 - 1.75 = -1.375$

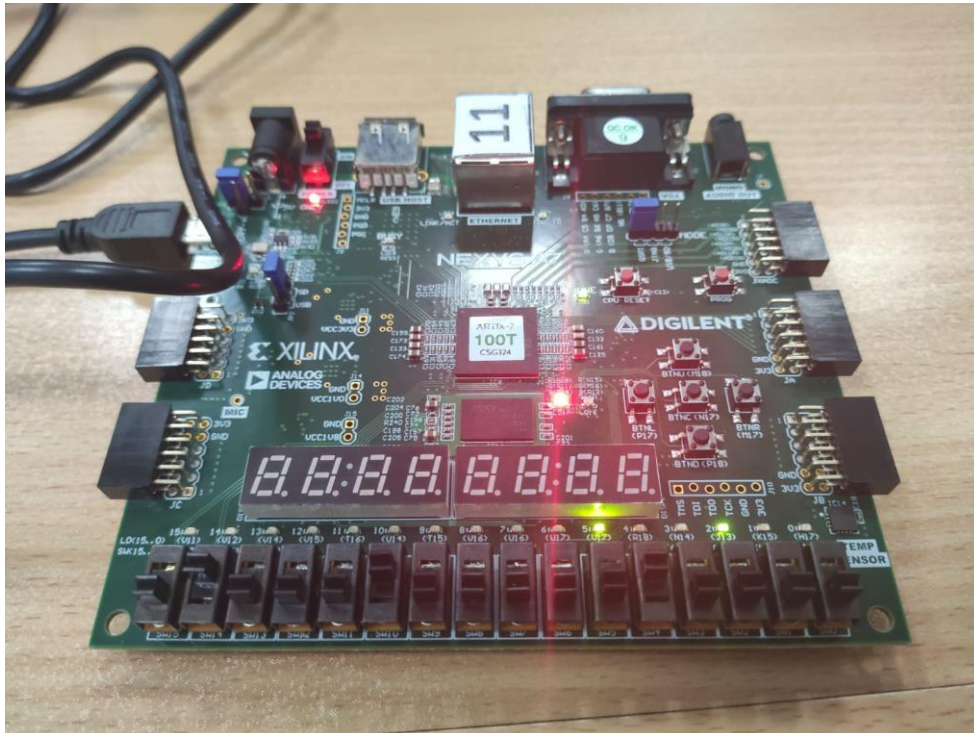


Şekil 3.6 Örnek 6- $0.375 + 1.75 = 2.125$

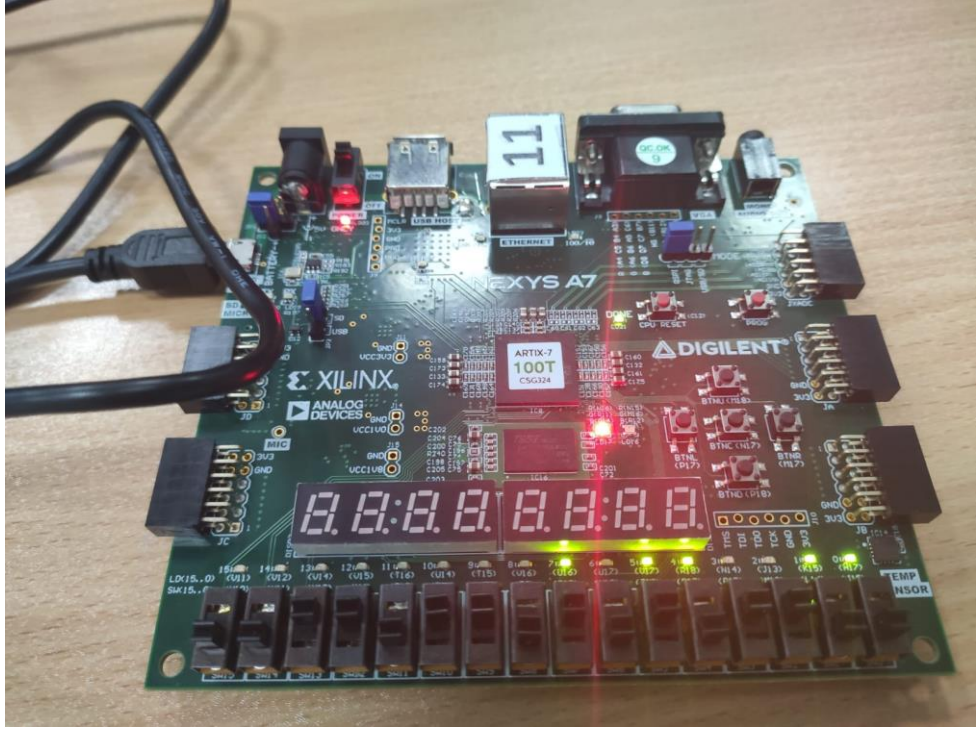
3.1.2 Çarpma Devresi



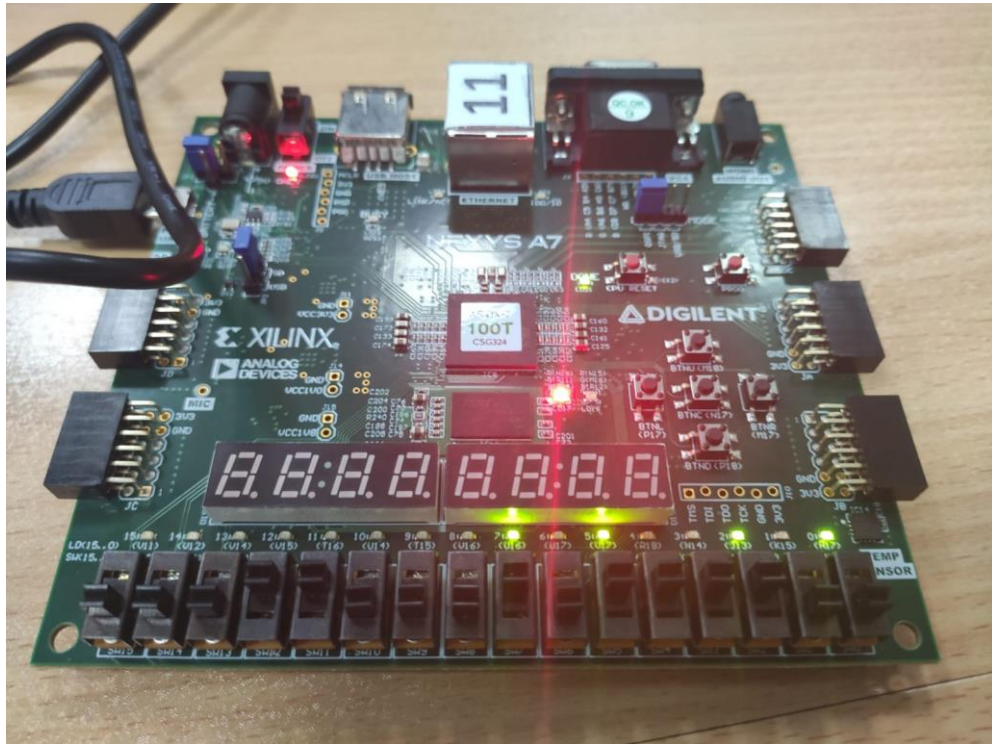
Şekil 3.7 Örnek 1- $-1.5 \times 2.5 = -3.75$



Şekil 3.8 Örnek 2- $2.5 \times 0.25 = 1$



Şekil 3.9 Örnek 3- $1.375 \times -0.875 = -1.203125$



Şekil 3.10 Örnek 4- $0.375 \times -1.75 = -0.65625$

- 1] A. B. Castillo, «DESIGN OF SINGLE PRECISION FLOAT ADDER (32-BIT NUMBERS) ACCORDING TO IEEE 754 STANDARD USING VHDL,» 25 Nisan 2015.
- [2] B. Taşkın, «FPGA Üzerinde Çarpma İşleminin Gerçeklenmesi,» [Çevrimiçi]. Available: <https://bertantaskin.com/floating-point-aritmetigi-ve-fpga-uzerinde-carpma-isleminin-gerceklestirilmesi/#:~:text=Floating%20Point%20Say%C4%B1larda%20%C3%87arpma%20%C4%B0%C5%9Fleminin%20Ger%C3%A7ekle%C5%9Ftirilmesi&text=G%C3%B6r%C3%BCld%C3%BC%C4%9F%C3%BC%2>. [Erişildi: Mayıs 2022].
- [3] T. Durmuş, «Kayan Nokta Gösterimi,» [Çevrimiçi]. Available: <https://talhadurmus.com/bilgisayar-bilimi/kayan-nokta-gosterimi>. [Erişildi: Mayıs 2022].
- [4] «Floating Point,» Vikipedi, [Çevrimiçi]. Available: https://tr.wikipedia.org/wiki/Kayan_nokta . [Erişildi: Mayıs 2022].