In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
gpu=pd.read_csv("gpu_specs_v6.csv")
```

In [3]:
```
gpu
```

Out[3]:

	manufacturer	productName	releaseYear	memSize	memBusWidth	gpuClock	memClock	un
0	NVIDIA	GeForce RTX 4050	2023.0	8.000	128.0	1925	2250.0	
1	Intel	Arc A350M	2022.0	4.000	64.0	300	1500.0	
2	Intel	Arc A370M	2022.0	4.000	64.0	300	1500.0	
3	Intel	Arc A380	2022.0	4.000	64.0	300	1500.0	
4	Intel	Arc A550M	2022.0	8.000	128.0	300	1500.0	
...	...	...	...	...	...	...	...	
2884	3dfx	Voodoo5 5000 AGP	NaN	0.016	128.0	166	166.0	
2885	3dfx	Voodoo5 5000 PCI	NaN	0.016	128.0	166	166.0	
2886	3dfx	Voodoo5 6000	NaN	0.032	128.0	166	166.0	
2887	Intel	Xe DG1	NaN	4.000	128.0	900	2133.0	
2888	Intel	Xe DG1-SDV	NaN	8.000	128.0	900	2133.0	

2889 rows × 16 columns

In [4]:
```
gpu.head()
```

Out[4]:

	manufacturer	productName	releaseYear	memSize	memBusWidth	gpuClock	memClock	unifie
0	NVIDIA	GeForce RTX 4050	2023.0	8.0	128.0	1925	2250.0	
1	Intel	Arc A350M	2022.0	4.0	64.0	300	1500.0	
2	Intel	Arc A370M	2022.0	4.0	64.0	300	1500.0	
3	Intel	Arc A380	2022.0	4.0	64.0	300	1500.0	
4	Intel	Arc A550M	2022.0	8.0	128.0	300	1500.0	

In [5]:
```
gpu.shape
```

Out[5]:
```
(2889, 16)
```

In [6]:
```
gpu.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2889 entries, 0 to 2888
Data columns (total 16 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   manufacturer   2889 non-null   object 
 1   productName    2889 non-null   object 
 2   releaseYear    2845 non-null   float64
 3   memSize        2477 non-null   float64
 4   memBusWidth    2477 non-null   float64
 5   gpuClock       2889 non-null   int64  
 6   memClock       2477 non-null   float64
 7   unifiedShader  2065 non-null   float64
 8   tmu            2889 non-null   int64  
 9   rop            2889 non-null   int64  
 10  pixelShader    824 non-null    float64
 11  vertexShader   824 non-null    float64
 12  igp            2889 non-null   object 
 13  bus            2889 non-null   object 
 14  memType        2889 non-null   object 
 15  gpuChip        2889 non-null   object 
dtypes: float64(7), int64(3), object(6)
memory usage: 361.2+ KB
```

In [7]:
```
gpu.rename(columns={"tmu":"Texture Mapping Unit"},inplace=True)
```

In [8]:
```
gpu.rename(columns={"rop":"Render output unit","igp":"integrated graphics process
```

In [9]:
```
gpu.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2889 entries, 0 to 2888
Data columns (total 16 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   manufacturer                  2889 non-null   object 
 1   productName                   2889 non-null   object 
 2   releaseYear                   2845 non-null   float64
 3   memSize                       2477 non-null   float64
 4   memBusWidth                   2477 non-null   float64
 5   gpuClock                      2889 non-null   int64  
 6   memClock                      2477 non-null   float64
 7   unifiedShader                 2065 non-null   float64
 8   Texture Mapping Unit          2889 non-null   int64  
 9   Render output unit            2889 non-null   int64  
 10  pixelShader                   824 non-null    float64
 11  vertexShader                  824 non-null    float64
 12  integrated graphics processor 2889 non-null   object 
 13  bus                           2889 non-null   object 
 14  memType                       2889 non-null   object 
 15  gpuChip                       2889 non-null   object 
dtypes: float64(7), int64(3), object(6)
memory usage: 361.2+ KB
```

In [10]:
```
gpu.isnull().sum()
```

Out[10]:
```
manufacturer                     0
productName                      0
releaseYear                     44
memSize                        412
memBusWidth                    412
gpuClock                         0
memClock                       412
unifiedShader                  824
Texture Mapping Unit             0
Render output unit               0
pixelShader                   2065
vertexShader                  2065
integrated graphics processor    0
bus                              0
memType                          0
gpuChip                          0
dtype: int64
```

In [11]:
```
gpu.drop(columns=["pixelShader","vertexShader"],inplace=True)
```

In [12]:
```
gpu.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2889 entries, 0 to 2888
Data columns (total 14 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   manufacturer                  2889 non-null   object 
 1   productName                   2889 non-null   object 
 2   releaseYear                   2845 non-null   float64
 3   memSize                       2477 non-null   float64
 4   memBusWidth                   2477 non-null   float64
 5   gpuClock                      2889 non-null   int64  
 6   memClock                      2477 non-null   float64
 7   unifiedShader                 2065 non-null   float64
 8   Texture Mapping Unit          2889 non-null   int64  
 9   Render output unit            2889 non-null   int64  
 10  integrated graphics processor  2889 non-null   object 
 11  bus                           2889 non-null   object 
 12  memType                       2889 non-null   object 
 13  gpuChip                       2889 non-null   object 
dtypes: float64(5), int64(3), object(6)
memory usage: 316.1+ KB
```

In [13]:
```
gpu.isnull().sum()
```

Out[13]:
```
manufacturer                     0
productName                      0
releaseYear                     44
memSize                        412
memBusWidth                    412
gpuClock                         0
memClock                       412
unifiedShader                  824
Texture Mapping Unit             0
Render output unit               0
integrated graphics processor    0
bus                              0
memType                          0
gpuChip                          0
dtype: int64
```

As memory size, memory bus width and memory clock specs are not available for the console based gpu and integrated gpu,therefore we will drop those rows because we only have to analyze the specifications of physical gpu units

In [14]:
```
gpu.dropna(inplace=True)
```

In [15]:
```
gpu.isnull().sum()
```

Out[15]:
```
manufacturer                     0
productName                      0
releaseYear                      0
memSize                          0
memBusWidth                      0
gpuClock                         0
memClock                         0
unifiedShader                    0
Texture Mapping Unit             0
Render output unit               0
integrated graphics processor    0
bus                              0
memType                          0
gpuChip                          0
dtype: int64
```

In [16]:
```
gpu.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1721 entries, 0 to 2340
Data columns (total 14 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   manufacturer                   1721 non-null   object 
 1   productName                    1721 non-null   object 
 2   releaseYear                    1721 non-null   float64
 3   memSize                        1721 non-null   float64
 4   memBusWidth                    1721 non-null   float64
 5   gpuClock                       1721 non-null   int64  
 6   memClock                       1721 non-null   float64
 7   unifiedShader                  1721 non-null   float64
 8   Texture Mapping Unit           1721 non-null   int64  
 9   Render output unit             1721 non-null   int64  
 10  integrated graphics processor  1721 non-null   object 
 11  bus                            1721 non-null   object 
 12  memType                        1721 non-null   object 
 13  gpuChip                        1721 non-null   object 
dtypes: float64(5), int64(3), object(6)
memory usage: 201.7+ KB
```

In [17]:
```
gpu["releaseYear"]=gpu["releaseYear"].astype("int64")
```

In [18]:
```
gpu["memBusWidth"]=gpu["memBusWidth"].astype("int64")
```

In [19]:
```
gpu["memClock"]=gpu["memClock"].astype("int64")
```

In [20]:
```
gpu["unifiedShader"]=gpu["unifiedShader"].astype("int64")
```

In [21]:
```
gpu.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1721 entries, 0 to 2340
Data columns (total 14 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   manufacturer                  1721 non-null   object
 1   productName                   1721 non-null   object
 2   releaseYear                   1721 non-null   int64
 3   memSize                       1721 non-null   float64
 4   memBusWidth                   1721 non-null   int64
 5   gpuClock                      1721 non-null   int64
 6   memClock                      1721 non-null   int64
 7   unifiedShader                 1721 non-null   int64
 8   Texture Mapping Unit          1721 non-null   int64
 9   Render output unit            1721 non-null   int64
 10  integrated graphics processor  1721 non-null   object
 11  bus                           1721 non-null   object
 12  memType                       1721 non-null   object
 13  gpuChip                       1721 non-null   object
dtypes: float64(1), int64(7), object(6)
memory usage: 201.7+ KB
```

In [22]:
```
gpu.isnull().sum()
```

Out[22]:
```
manufacturer                    0
productName                     0
releaseYear                     0
memSize                         0
memBusWidth                     0
gpuClock                        0
memClock                        0
unifiedShader                   0
Texture Mapping Unit            0
Render output unit              0
integrated graphics processor   0
bus                             0
memType                         0
gpuChip                         0
dtype: int64
```

```
In [23]: gpu
```

Out[23]:

	manufacturer	productName	releaseYear	memSize	memBusWidth	gpuClock	memClock	un
0	NVIDIA	GeForce RTX 4050	2023	8.000	128	1925	2250	
1	Intel	Arc A350M	2022	4.000	64	300	1500	
2	Intel	Arc A370M	2022	4.000	64	300	1500	
3	Intel	Arc A380	2022	4.000	64	300	1500	
4	Intel	Arc A550M	2022	8.000	128	300	1500	
...	...	...	...	...	...	...	...	
2113	NVIDIA	Tesla S870	2007	1.536	384	600	800	
2114	ATI	Xbox 360 GPU 80nm	2007	0.512	128	500	700	
2157	NVIDIA	GeForce 8800 GTS 640	2006	0.640	320	513	792	
2158	NVIDIA	GeForce 8800 GTX	2006	0.768	384	576	900	
2340	ATI	Xbox 360 GPU 90nm	2005	0.512	128	500	700	

1721 rows × 14 columns

In [24]:
```
gpu.describe()
```

Out[24]:

	releaseYear	memSize	memBusWidth	gpuClock	memClock	unifiedShader	Tex Map
count	1721.000000	1721.000000	1721.000000	1721.000000	1721.000000	1721.000000	1721.00
mean	2013.571761	4.345259	322.584544	861.147008	1102.266124	1170.514817	71.45
std	4.130951	8.273479	761.177359	325.848376	407.848886	1769.078898	84.93
min	2005.000000	0.128000	32.000000	300.000000	266.000000	8.000000	4.00
25%	2010.000000	1.024000	128.000000	620.000000	800.000000	160.000000	20.00
50%	2013.000000	2.000000	128.000000	796.000000	1000.000000	480.000000	40.00
75%	2017.000000	4.000000	256.000000	1005.000000	1375.000000	1536.000000	96.00
max	2023.000000	128.000000	8192.000000	2331.000000	2257.000000	17408.000000	880.00

In [25]:
```
#1.Which gpu were released in year 2013?
```

In [100]:
```
a=gpu[(gpu["releaseYear"]>=2010)&(gpu["releaseYear"]<=2015)]
a
```

Out[100]:

	manufacturer	productName	releaseYear	memSize	memBusWidth	gpuClock	memClock	un
649	AMD	FirePro S9170	2015	32.000	512	930	1250	
650	AMD	FirePro W4130M	2015	1.024	128	775	1000	
651	AMD	FirePro W4150M	2015	1.024	128	800	1000	
652	AMD	FirePro W4170M	2015	2.000	128	825	1000	
653	AMD	FirePro W4190M	2015	2.000	128	825	1000	
...	...	...	...	...	...	...	...	
1673	AMD	Radeon HD 6850	2010	1.024	256	775	1000	
1674	AMD	Radeon HD 6870	2010	1.024	256	900	1050	
1675	AMD	Radeon HD 6950	2010	2.000	256	800	1250	
1676	AMD	Radeon HD 6970	2010	2.000	256	880	1375	
1678	ATI	Xbox 360 S GPU	2010	0.512	128	500	700	

866 rows × 14 columns

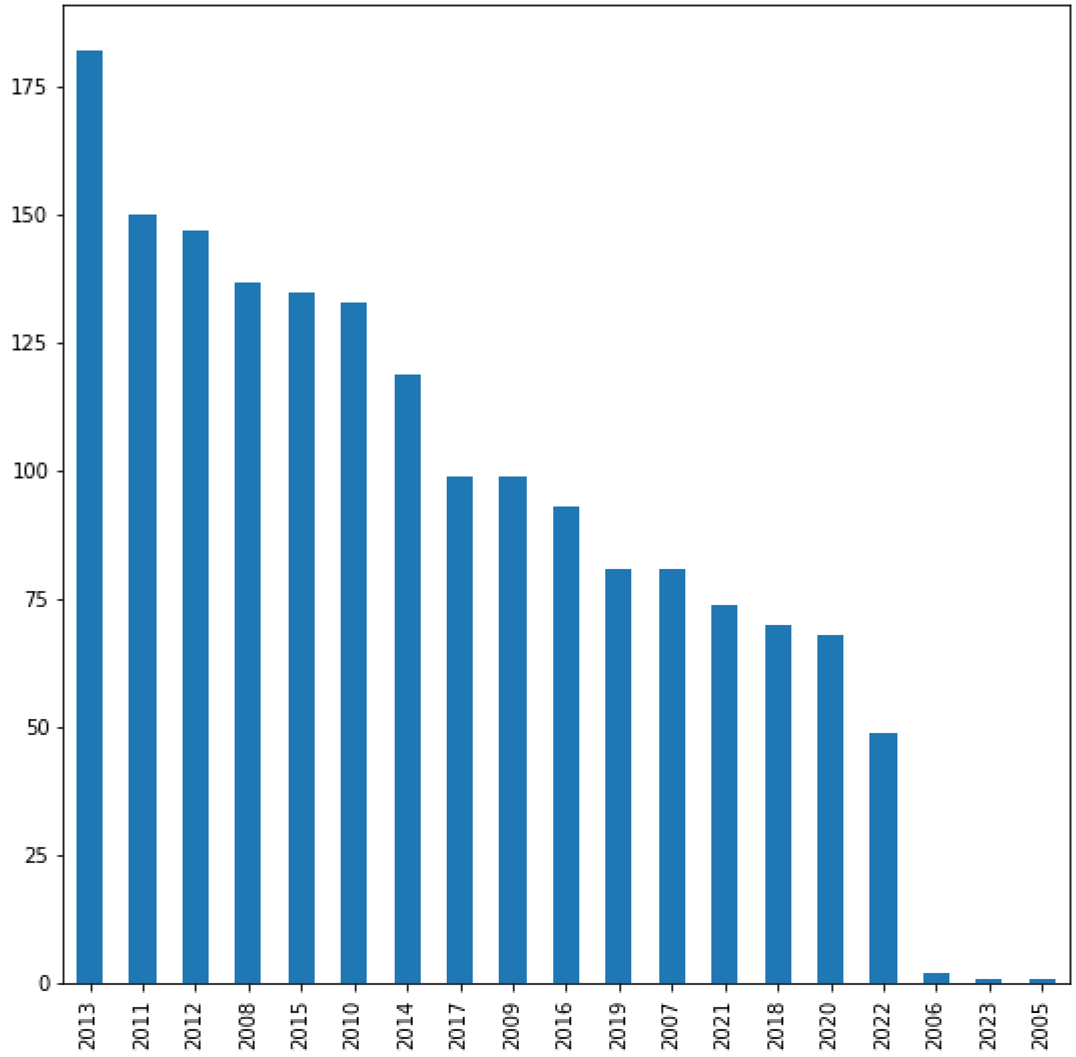
In [27]:
```
#2.Which gpu has the highest memory clock?
```

In [38]:
```
gpu[gpu["memClock"]==gpu["memClock"].max()]["productName"]
```

Out[38]:
```
437    GeForce GTX 1060 6 GB 9Gbps
Name: productName, dtype: object
```

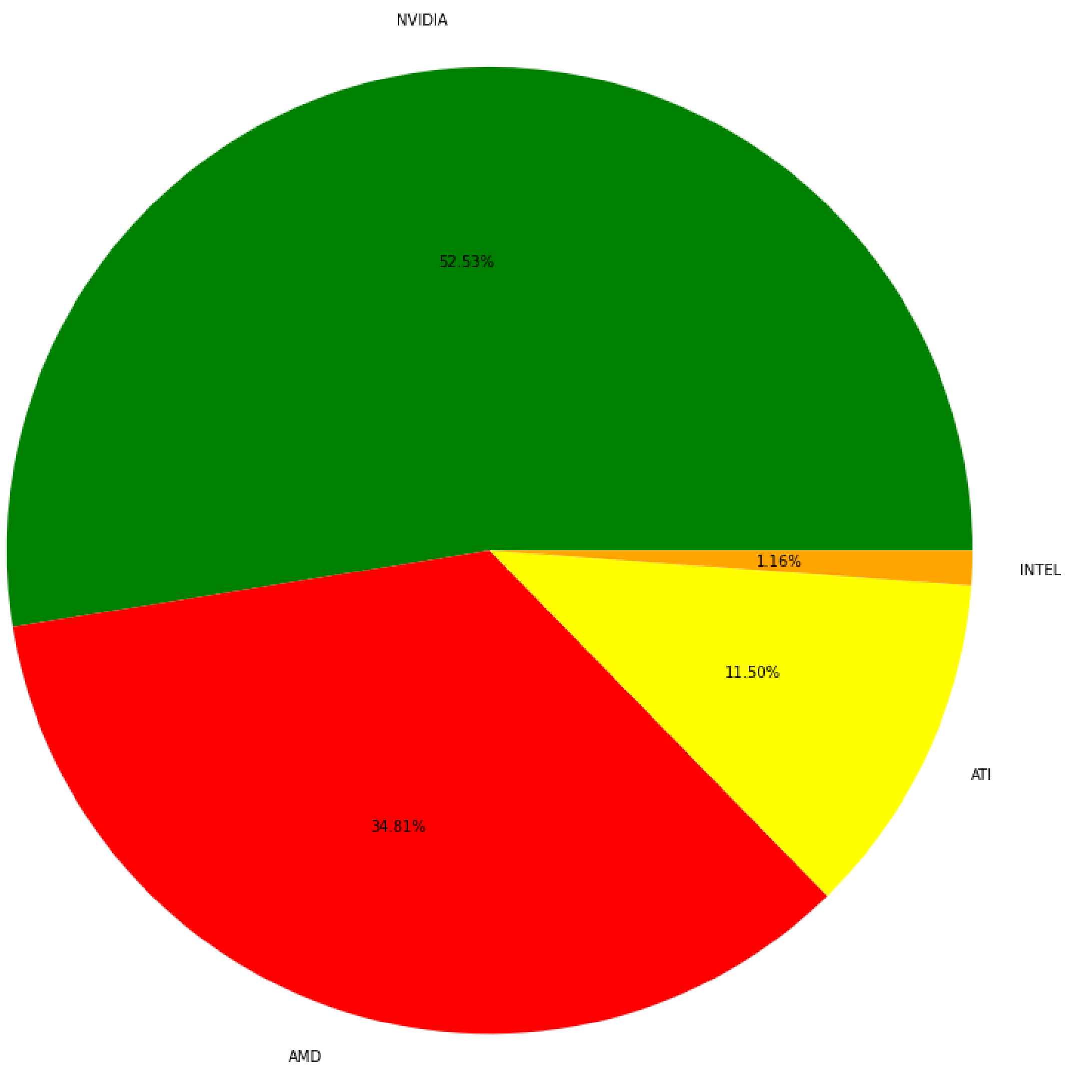
In [29]:
```
#3.Highest number of gpu released in a year?
plt.figure(figsize=(9,9))
gpu["releaseYear"].value_counts().plot(kind="bar")
```

Out[29]:
```
<AxesSubplot: >
```



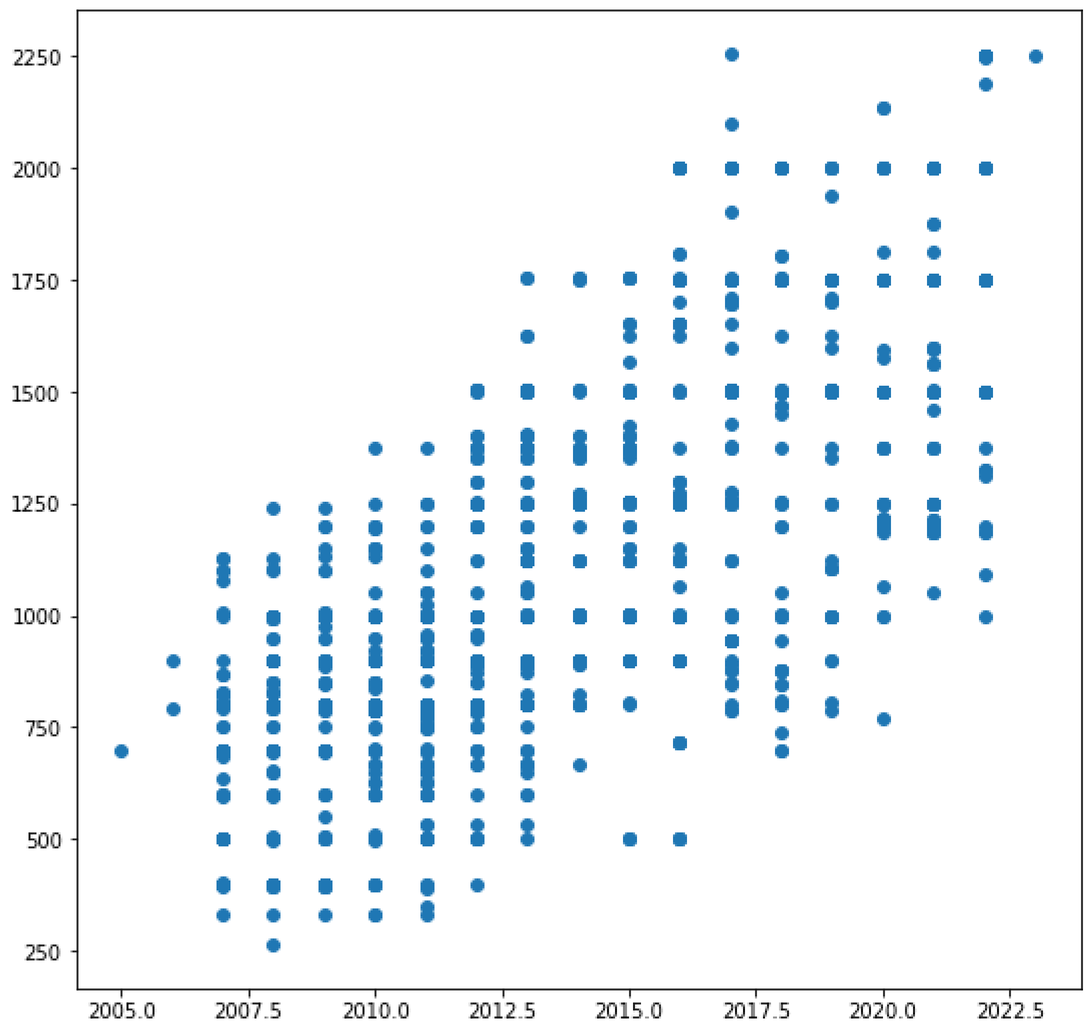
```
The highest number of gpu were released in year 2013 which is 182 which 2005
has the lowest release of only 1 gpu
```

```
In [101]: #4.Which manufacturer has more percentage of gpu models?
          y=gpu["manufacturer"].value_counts()
          plt.figure(figsize=(15,15))
          plt.pie(y,labels=["NVIDIA","AMD","ATI","INTEL"],colors=["green","red","yellow","c
          plt.show()
```



```
Nvidia has more than 50% of market share of gpu which AMD has 34.81 percent of
the share. while intel has lowest share of only 1.16%
```

In [36]:
```
#5.Difference in memory clock speed with respect to year
plt.figure(figsize=(9,9))
plt.scatter(gpu["releaseYear"],gpu["memClock"])
plt.show()
```



```
As the technology advanced,gpu were provided with better and faster memory
clock speeds which helps in faster rendering of graphics.
```

In [33]:
```
#6.How many gpu has PCIe4.0x16 slots?
```

In [48]:
```
a=gpu[gpu["bus"]=="PCIe 4.0 x16"]
a["bus"].value_counts()
```

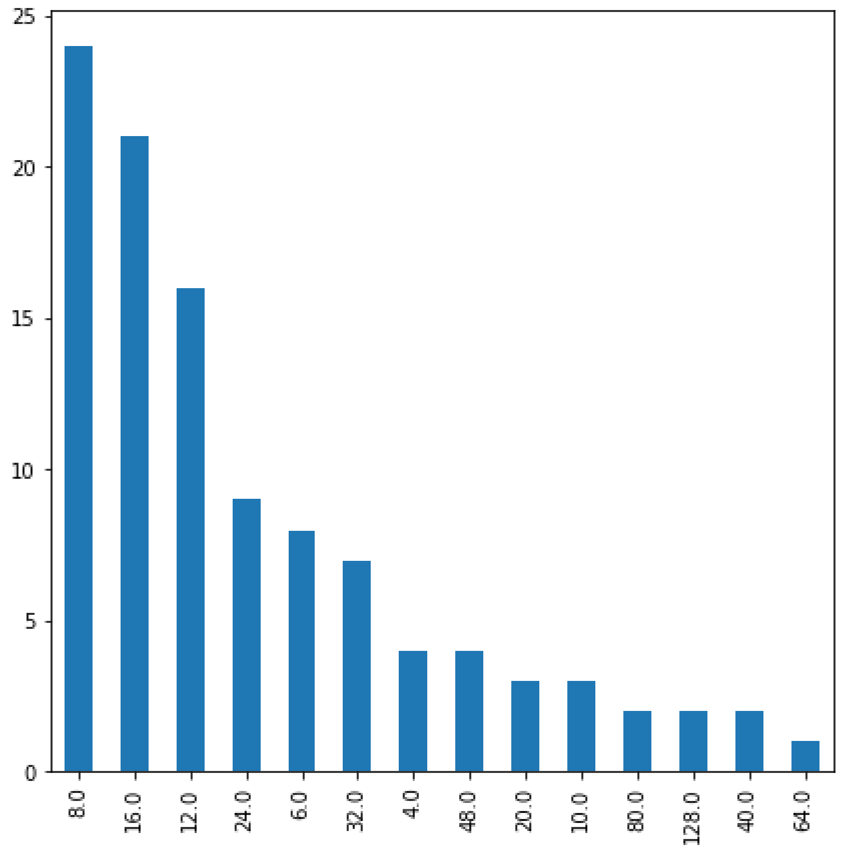
Out[48]:
```
PCIe 4.0 x16    106
Name: bus, dtype: int64
```

```
In [52]: #7. Most Common memory type
         gpu["memType"].value_counts().head(1)
```

```
Out[52]: GDDR5    712
         Name: memType, dtype: int64
```

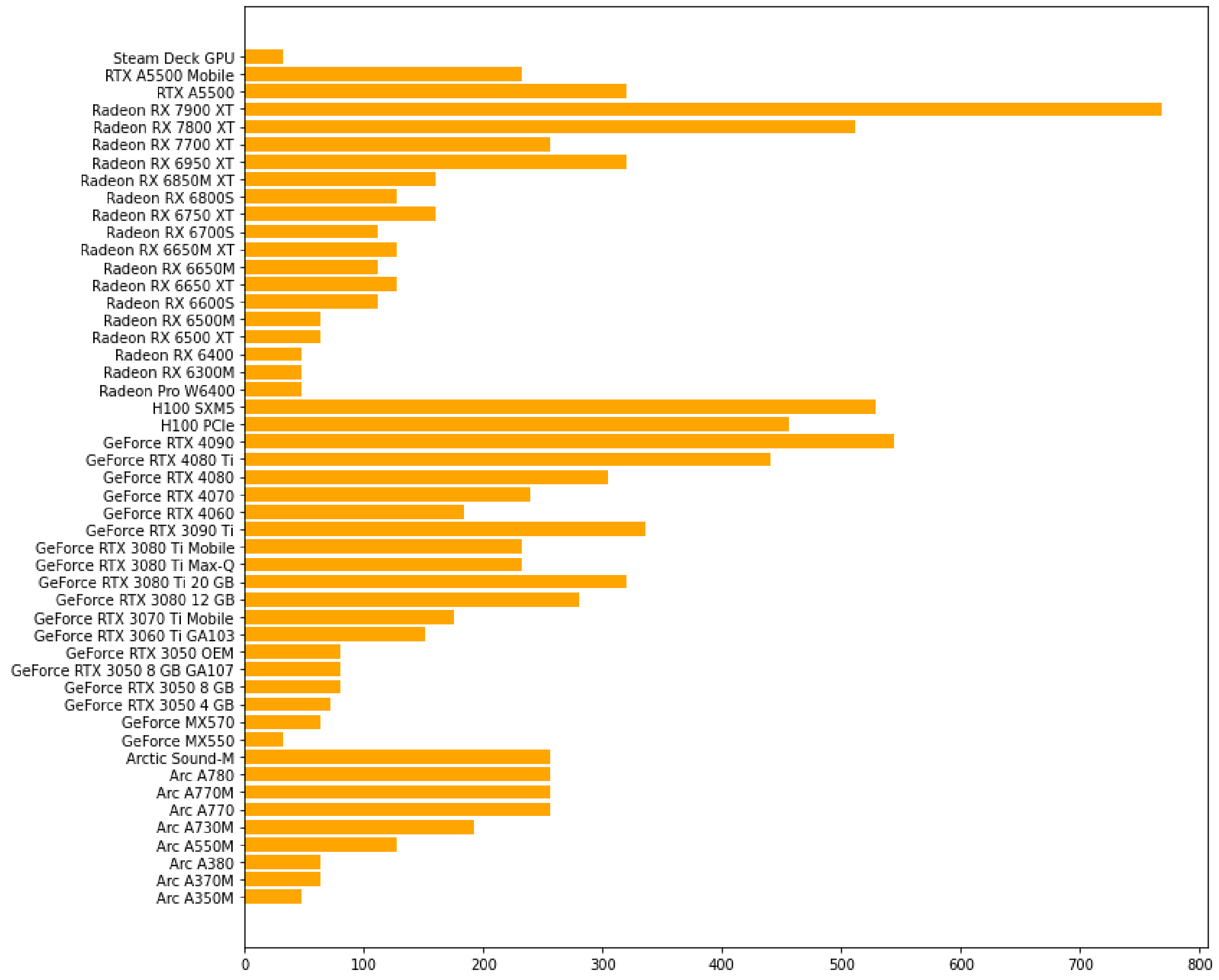
```
In [63]: #8.What is the most common memory size for PCIe 4.0x16?
         plt.figure(figsize=(7,7))
         a["memSize"].value_counts().plot(kind="bar")
```

```
Out[63]: <AxesSubplot: >
```



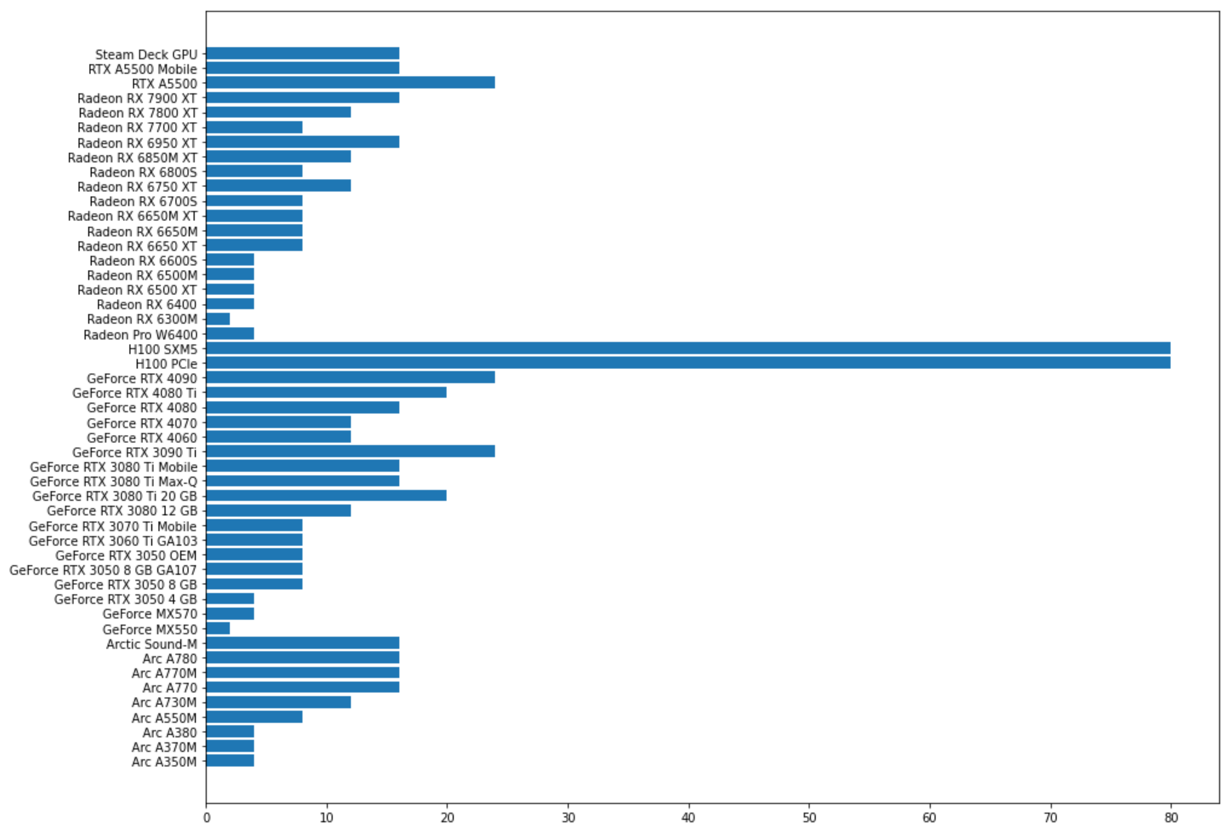
```
Most common memory size is 8gb for PCIe 4.0x16 slot which is 24
```

In [93]:
```
#9. Which gpu in year 2020 has the highestTexture Mapping Unit?
a=gpu[(gpu["releaseYear"]==2022)]
plt.figure(figsize=(12,12))
plt.barh(a["productName"],a["Texture Mapping Unit"],color="orange")
plt.show()
```



As we can see from the above bar plot Radeon RX 7900XT has the highest texture mapping unit released in the year 2020 while the lowest are steam deck gpu and Geforce MX550

In [120]:
```
#10. Highest memory size in year 2022
m=gpu[gpu["releaseYear"]==2022]
y=m["memSize"]
x=m["productName"]
plt.figure(figsize=(15,12))
plt.barh(x,y,label="gpu CLock")
plt.show()
```



As we can see highest memory size is 80gb which are H100SXM5 and H100PCIe.
These gpu are only available for enterprise and data server market and are not available for common consumer.
Considering the gpus available for common consumer highest memory size in 2022 is 24gb while lowest is 2gb

In [124]:
```
#11. Maximum memory size
gpu["memSize"].max()
```

Out[124]:
```
128.0
```