In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
data1 = pd.read_csv("amd_processors.csv")
data2 = pd.read_csv("intel_processors.csv")
data3 = pd.read_csv("cpu_benchmarks.csv")
```

In [3]:

```
data1.head()
```

Out[3]:

	id	cores	threads	name	launch_date	lithography	base_frequency	turbo_frequency	cache_
0	1	16.0	32.0	AMD Ryzen 9 5950X	05-11-2020	7.0	3400	4900.0	Na
1	2	12.0	24.0	AMD Ryzen 9 5900X	05-11-2020	7.0	3700	4800.0	Na
2	3	12.0	24.0	AMD Ryzen 9 5900 (OEM Only)	12-01-2021	7.0	3000	4700.0	Na
3	4	8.0	16.0	AMD Ryzen 7 5800X	05-11-2020	7.0	3800	4700.0	Na
4	5	8.0	16.0	AMD Ryzen 7 5800 (OEM Only)	12-01-2021	7.0	3400	4600.0	Na

In [4]:

```
data2.head()
```

Out[4]:

	id	cores	threads	name	processor_number	launch_date	lithography	bus_speed	base_fre
0	1	2	2.0	Intel Atom C2338	C2338	01-08-2013	22.0	NaN	
1	2	2	2.0	Intel Atom C2350	C2350	01-08-2013	22.0	NaN	
2	3	2	2.0	Intel Atom C2358	C2358	01-08-2013	22.0	NaN	
3	4	4	4.0	Intel Atom C2518	C2518	01-08-2013	22.0	NaN	
4	5	4	8.0	Intel Core i7-8709G	i7-8709G	01-02-2018	14.0	8.0	

5 rows × 26 columns

In [5]:

```
data3.head()
```

Out[5]:

	id	manufacturer	sku	rating	url
0	1	amd	10456	46113	https://www.cpubenchmark.net/cpu.php?cpu=AMD+R...
1	2	amd	10461	39634	https://www.cpubenchmark.net/cpu.php?cpu=AMD+R...
2	3	amd	10466	28481	https://www.cpubenchmark.net/cpu.php?cpu=AMD+R...
3	4	amd	10471	22165	https://www.cpubenchmark.net/cpu.php?cpu=AMD+R...
4	5	amd	9931	32902	https://www.cpubenchmark.net/cpu.php?cpu=AMD+R...

In [6]:

```
data1.shape
```

Out[6]:

```
(538, 20)
```

In [7]:

```
data2.shape
```

Out[7]:

```
(1098, 26)
```

In [8]:

```
data3.shape
```

Out[8]:

```
(1667, 5)
```

In [9]:

```
data1.columns
```

Out[9]:

```
Index(['id', 'cores', 'threads', 'name', 'launch_date', 'lithography',
       'base_frequency', 'turbo_frequency', 'cache_l1', 'cache_l2', 'cache
_l3',
       'tdp', 'product_line', 'socket', 'memory_type', 'url',
       'vertical_segment', 'max_temp', 'max_memory_speed', 'sku'],
      dtype='object')
```

In [10]:

```
data2.columns
```

Out[10]:

```
Index(['id', 'cores', 'threads', 'name', 'processor_number', 'launch_dat
e',
       'lithography', 'bus_speed', 'base_frequency', 'turbo_frequency',
       'configurable_tdp_up_frequency', 'cache_size', 'tdp',
       'configurable_tdp_up', 'price', 'product_line', 'socket', 'memory_t
ype',
       'url', 'vertical_segment', 'max_memory_size', 'status', 'max_temp',
       'sku', 'package_size', 'fullname'],
      dtype='object')
```

In [11]:

```
data3.columns
```

Out[11]:

```
Index(['id', 'manufacturer', 'sku', 'rating', 'url'], dtype='object')
```

In [12]:

```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 538 entries, 0 to 537
Data columns (total 20 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   id                538 non-null    int64  
 1   cores             533 non-null    float64
 2   threads           478 non-null    float64
 3   name              538 non-null    object 
 4   launch_date       185 non-null    object 
 5   lithography       440 non-null    float64
 6   base_frequency    538 non-null    int64  
 7   turbo_frequency   457 non-null    float64
 8   cache_l1          285 non-null    float64
 9   cache_l2          476 non-null    float64
 10  cache_l3          331 non-null    float64
 11  tdp               523 non-null    float64
 12  product_line      538 non-null    object 
 13  socket            520 non-null    object 
 14  memory_type       503 non-null    object 
 15  url               538 non-null    object 
 16  vertical_segment  538 non-null    object 
 17  max_temp          407 non-null    float64
 18  max_memory_speed  442 non-null    float64
 19  sku               538 non-null    int64  
dtypes: float64(10), int64(3), object(7)
memory usage: 84.2+ KB
```

In [13]:

```
data2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1098 entries, 0 to 1097
Data columns (total 26 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              1098 non-null   int64
 1   cores                           1098 non-null   int64
 2   threads                         1094 non-null   float64
 3   name                            1098 non-null   object
 4   processor_number                1098 non-null   object
 5   launch_date                     1098 non-null   object
 6   lithography                     1089 non-null   float64
 7   bus_speed                       713 non-null    float64
 8   base_frequency                  1055 non-null   float64
 9   turbo_frequency                 871 non-null    float64
 10  configurable_tdp_up_frequency   98 non-null     float64
 11  cache_size                      1098 non-null   int64
 12  tdp                             1044 non-null   float64
 13  configurable_tdp_up             102 non-null    float64
 14  price                           839 non-null    float64
 15  product_line                    1098 non-null   object
 16  socket                          1046 non-null   object
 17  memory_type                     1094 non-null   object
 18  url                             1098 non-null   object
 19  vertical_segment                1098 non-null   object
 20  max_memory_size                 1092 non-null   float64
 21  status                          1098 non-null   object
 22  max_temp                        997 non-null    float64
 23  sku                             1098 non-null   int64
 24  package_size                    1048 non-null   object
 25  fullname                        1098 non-null   object
dtypes: float64(11), int64(4), object(11)
memory usage: 223.2+ KB
```

In [14]:

```
data3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1667 entries, 0 to 1666
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   id            1667 non-null   int64
 1   manufacturer  1667 non-null   object
 2   sku           1667 non-null   int64
 3   rating        1667 non-null   int64
 4   url           1667 non-null   object
dtypes: int64(3), object(2)
memory usage: 65.2+ KB
```

In [15]:

```
data1.describe()
```

Out[15]:

	id	cores	threads	lithography	base_frequency	turbo_frequency	
count	538.000000	533.000000	478.000000	440.000000	538.000000	457.000000	2
mean	269.500000	8.221388	14.845188	21.454545	2859.211896	3684.463895	18
std	155.451493	10.689384	22.932795	10.181171	742.696634	581.104974	164
min	1.000000	2.000000	2.000000	7.000000	56.000000	1500.000000	
25%	135.250000	4.000000	4.000000	12.000000	2300.000000	3350.000000	2
50%	269.500000	4.000000	8.000000	28.000000	3000.000000	3800.000000	3
75%	403.750000	8.000000	16.000000	28.000000	3500.000000	4100.000000	5
max	538.000000	64.000000	128.000000	65.000000	4700.000000	5000.000000	1966

In [16]:

```
data2.describe()
```

Out[16]:

	id	cores	threads	lithography	bus_speed	base_frequency	turbo
count	1098.000000	1098.000000	1094.000000	1089.000000	713.000000	1055.00000	
mean	549.500000	7.590164	14.213894	15.052342	6.417251	2574.64455	3
std	317.109603	7.403193	15.188295	3.725658	1.831803	731.75703	
min	1.000000	2.000000	2.000000	10.000000	0.000000	700.00000	1
25%	275.250000	2.000000	4.000000	14.000000	5.000000	2100.00000	3
50%	549.500000	4.000000	8.000000	14.000000	8.000000	2500.00000	3
75%	823.750000	8.000000	16.000000	14.000000	8.000000	3100.00000	4
max	1098.000000	56.000000	112.000000	32.000000	8.000000	4300.00000	5

In [17]:

```
data3.describe()
```

Out[17]:

	id	sku	rating
count	1667.000000	1667.000000	1667.000000
mean	834.000000	84215.235753	8068.811038
std	481.365765	60703.514320	10191.251437
min	1.000000	1221.000000	106.000000
25%	417.500000	42808.000000	2017.000000
50%	834.000000	76622.000000	4522.000000
75%	1250.500000	97472.500000	9738.500000
max	1667.000000	217187.000000	87767.000000

In [18]:

```
data1.isnull().sum()
```

Out[18]:

```
id                   0
cores                5
threads             60
name                 0
launch_date        353
lithography         98
base_frequency       0
turbo_frequency     81
cache_l1           253
cache_l2            62
cache_l3           207
tdp                 15
product_line         0
socket              18
memory_type         35
url                  0
vertical_segment     0
max_temp           131
max_memory_speed    96
sku                  0
dtype: int64
```

In [19]:

```
data2.isnull().sum()
```

Out[19]:

```
id                                  0
cores                               0
threads                             4
name                                0
processor_number                    0
launch_date                         0
lithography                         9
bus_speed                         385
base_frequency                     43
turbo_frequency                   227
configurable_tdp_up_frequency    1000
cache_size                          0
tdp                                54
configurable_tdp_up               996
price                             259
product_line                        0
socket                             52
memory_type                         4
url                                 0
vertical_segment                    0
max_memory_size                     6
status                              0
max_temp                          101
sku                                 0
package_size                       50
fullname                            0
dtype: int64
```

In [20]:

```
data3.isnull().sum()
```

Out[20]:

```
id              0
manufacturer    0
sku             0
rating          0
url             0
dtype: int64
```

In [21]:

```
data1.nunique()
```

Out[21]:

```
id                  538
cores                13
threads              15
name                525
launch_date          36
lithography           7
base_frequency       48
turbo_frequency      36
cache_l1             24
cache_l2             11
cache_l3             10
tdp                  35
product_line        127
socket               25
memory_type           9
url                 538
vertical_segment      6
max_temp             31
max_memory_speed     19
sku                 538
dtype: int64
```

In [22]:

```
data1.cores.unique()
```

Out[22]:

```
array([16., 12.,  8.,  6., 64., 32., 24.,  4.,  2., 56., 48., 28., nan,
        3.])
```

In [23]:

```
data1.base_frequency.unique()
```

Out[23]:

```
array([3400, 3700, 3000, 3800, 3900, 2700, 3500, 4000, 2900, 3300, 3200,
       2800, 1900, 1800, 3100, 2300, 2100, 2600, 3600, 2000, 2200, 2400,
       1400, 1200, 2500, 1700, 1600,  200, 4100, 2450, 2950, 2750, 2850,
       2650, 2250, 2350, 1550, 1500, 4700, 4400, 4200,   56, 2050, 1300,
       1450, 1650, 1350, 1000], dtype=int64)
```

In [24]:

```
data1.turbo_frequency.unique()
```

Out[24]:

```
array([4900., 4800., 4700., 4600., 4500., 4200., 4300., 4400., 4000.,
       3800., 3900., 4100., 3700., 3500.,   nan, 3300., 3200., 2800.,
       2600., 2300., 3400., 3600., 2900., 3100., 2700., 2500., 2400.,
       2200., 3000., 3675., 3650., 3450., 3350., 5000., 2000., 1800.,
       1500.])
```

In [25]:

```
data1.memory_type.unique()
```

Out[25]:

```
array(['DDR4', 'DDR4 - Up to 3200MHz, LPDDR4 - Up to 4266MHz', 'LPDDR4',
       nan, 'DDR4, LPDDR4', 'DDR3', 'DDR3/DDR3L', 'Not Listed', 'DDR3L',
       'DDR2'], dtype=object)
```

In [26]:

```
data1.max_memory_speed.unique()
```

Out[26]:

```
array([3200.,   nan, 2400., 4267., 2933., 2667., 1600., 2133., 1866.,
        186., 2666., 1333., 1865., 1599.,  800., 2000., 1800., 2200.,
       1400.,  667.])
```

In [28]:

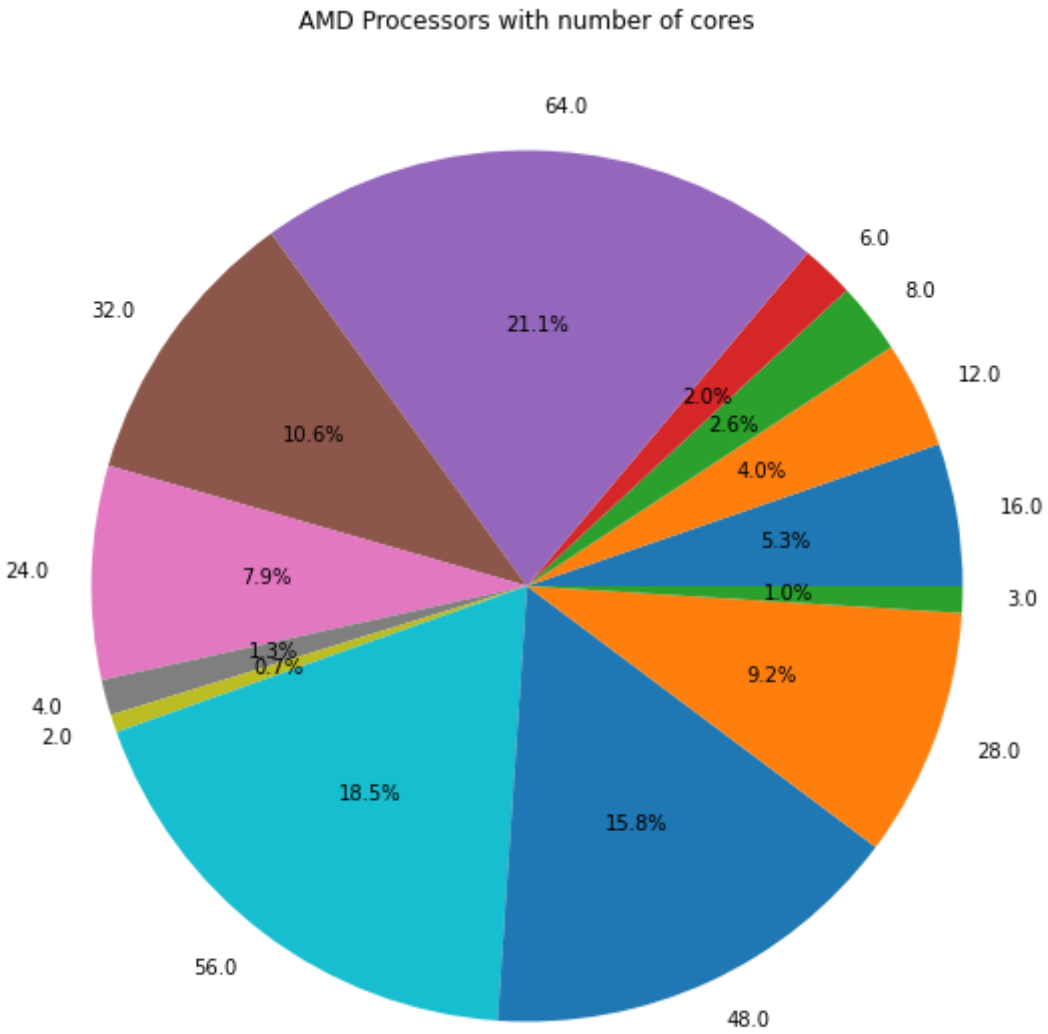
```
data1.drop(['id','url'] ,axis=1 , inplace =True)
```

In [29]:

```
amd_cores = list(data1['cores'].dropna().unique())
```

In [33]:

```
plt.figure(figsize =(15, 10))
plt.pie(amd_cores, labels = amd_cores , autopct='%1.1f%%')
plt.title('AMD Processors with number of cores')
plt.show()
```



In [35]:

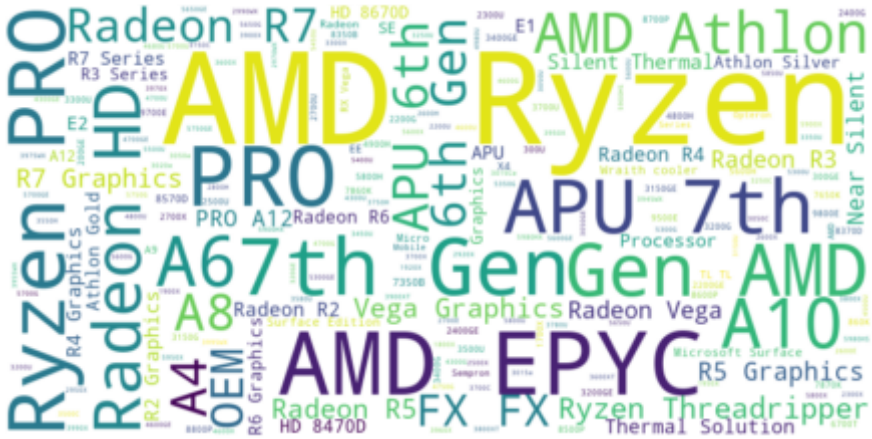
```
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
```

In [36]:

```
text = " ".join(review for review in data1.name.astype(str))
stopwords = set(STOPWORDS)
```

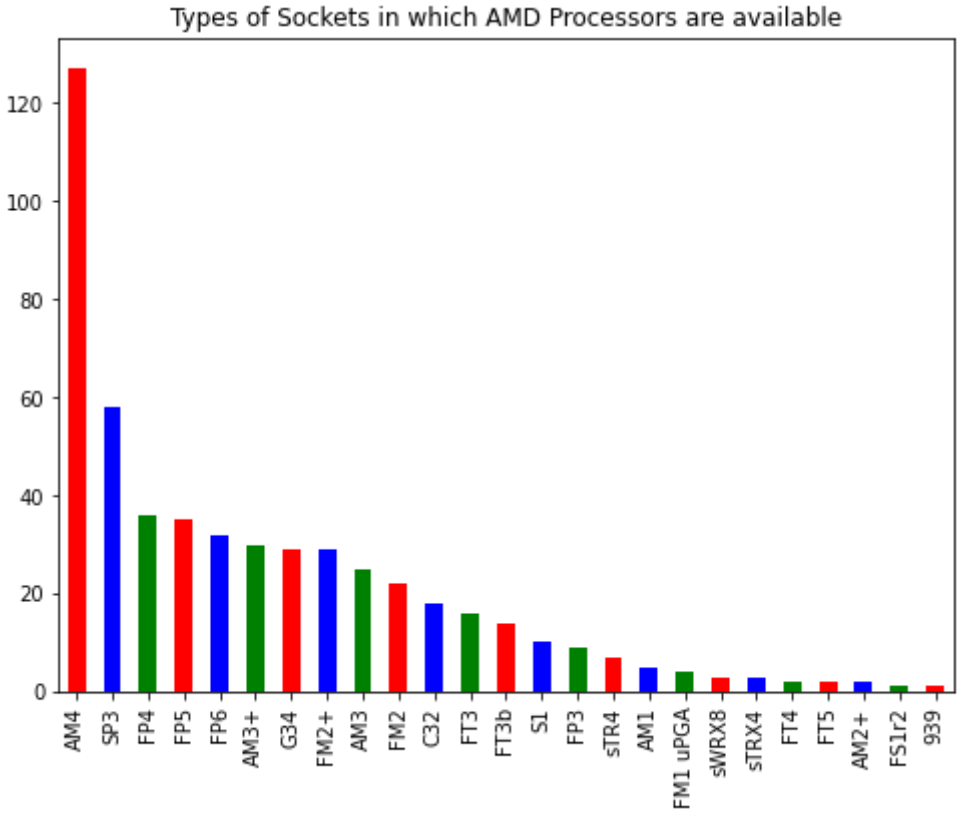
In [37]:

```
wordcloud = WordCloud(stopwords=stopwords, background_color="white", width=800, height=4
plt.axis("off")
plt.tight_layout(pad=0)
plt.imshow(wordcloud, interpolation='bilinear')
plt.show()
```



In [38]:

```
plt.figure(figsize =(8, 6))
data1['socket'].value_counts().plot.bar(color=['red','blue','green'])
plt.title('Types of Sockets in which AMD Processors are available')
plt.show()
```



In [39]:

```
data2.nunique()
```

Out[39]:

```
id                                1098
cores                               18
threads                             20
name                              1090
processor_number                  1087
launch_date                         40
lithography                          4
bus_speed                            6
base_frequency                      46
turbo_frequency                     38
configurable_tdp_up_frequency       24
cache_size                          43
tdp                                 96
configurable_tdp_up                  9
price                              322
product_line                        42
socket                              36
memory_type                         77
url                               1098
vertical_segment                     5
max_memory_size                     20
status                               3
max_temp                            52
sku                               1098
package_size                        48
fullname                          1098
dtype: int64
```

In [40]:

```
data2.cores.unique()
```

Out[40]:

```
array([ 2,  4,  8, 16, 14, 20, 12, 18,  6, 22, 10, 24, 28, 26, 56, 32, 36,
       38], dtype=int64)
```

In [41]:

```
data2.base_frequency.unique()
```

Out[41]:

```
array([1700., 3100., 3400., 2600., 2400., 2000., 2200., 2700., 2800.,
       3500., 2500., 1300., 3000., 1500., 2300., 3200., 2900., 1400.,
       1600., 1900., 1800., 3300., 2100., 3900., 3600., 4000., 2410.,
       1200., 1460., 1860., 2130., 1830., 2160., 1580., 3800., 3700.,
       4200., 1040., 1100., 2260., 1250., 1000.,  800.,  700., 4100.,
         nan, 4300.])
```

In [42]:

```
data2.turbo_frequency.unique()
```

Out[42]:

```
array([2000.,   nan, 4100., 4000., 3600., 3400., 3000., 3100., 3200.,
       3800., 3900., 3700., 2600., 1900., 3300., 2900., 2300., 3500.,
       2700., 4200., 2800., 4400., 2500., 4300., 4500., 4700., 4600.,
       2400., 5000., 2200., 4800., 4900., 2100., 1700., 1600., 5100.,
       5300., 5200., 1300.])
```

In [43]:

```
data2.memory_type.unique()
```

Out[43]:

```
array(['DDR3/DDR3L 1333', 'DDR4-2400',
       'DDR4-1866/2133, DDR3L-1333/1600 @ 1.35V', 'DDR3L 1333/1600',
       'DDR3L 1333/1600, LPDDR3 1600/1866',
       'DDR3L 1333/1600, LPDDR3 1333/1600', 'DDR3L 1600',
       'DDR3-1333/1600, DDR3L-1333/1600 @ 1.5V',
       'DDR3 and DDR3L 1333/1600 at 1.5V', 'DDR3 1333/1600',
       'DDR3L-1333, 1600; LPDDR3-1333, 1600',
       'DDR3L 1600/1866 LPDDR3 1600/1866', 'DDR3L-1333/1600/1866 @ 1.5V',
       'DDR4-2666', 'DDR3L-1333/1600 @ 1.5V',
       'DDR4-2133, LPDDR3-1866, DDR3L-1600',
       'DDR4-2133, LPDDR3 - 1866, DDR3L-1600', 'DDR3L 1333', 'DDR3L-1600',
       'DDR3L/LPDDR3 up to 1866 MT/s; LPDDR4 up to 2400 MT/s',
       'DDR4/LPDDR4 upto 2400 MT/s', 'LPDDR3-1866, DDR3L-1600',
       'DDR4-2400, LPDDR3-2133, DDR3L-1600',
       'DDR4-2133/2400, DDR3L-1333/1600 @ 1.35V', '2667', 'DDR4-2933',
       '4x32 LPDDR4/x 3733MT/s Max (8GB, 16GB @3200MT/s) / 2x64 DDR4 3200M
T/s Max 32GB',
       'DDR3L 1066', 'DDR4-2400, LPDDR3-2133', 'DDR4-2666, LPDDR3-2133',
       'DDR4-2667', 'DDR4/LPDDR4 upto 2400MT/s',
       'DDR4/LPDDR4 (SRET0) up to 2400 MT/s DDR4 (SRKLL) up to 2400 MT/s',
       'DDR4/LPDDR4 up to 2400 MT/s',
       '4x32 LPDDR4/x 3200MT/s Max 16GB / 2x64 DDR4 3200MT/s Max 32GB',
       'DDR4 LPDDR4x', 'LPDDR3-2133, DDR3L-1600',
       'DDR3L 1333/1600  LPDDR 1333 /1600', 'DDR3 1066', 'DDR4 RDIMM',
       'DDR3L-1600, DDR4-2400', nan, 'DDR3/DDR3L 1600', 'DDR3 1333',
       'DDR 1333', 'DDR4: 1866', 'LPDDR4-3733',
       'DDR3-1333, DDR3L-1333 @ 1.5V', 'DDR4-3200, LPDDR4-3733',
       'DDR4-2666, LPDDR3-2133, LPDDR4-2933', 'DDR4-3200', 'RDIMM',
       'DDR4-2400, LPDDR4/x-2400', 'DDR4: 2400', 'DDR4: 2133',
       'DDR4-1866', 'DDR3 1066/1333/1600', 'DDR3/L/-RS 1333/1600',
       'DDR3 1333/1600/1866', 'DDR4 1600/1866/2133', 'DDR4 2400/2133',
       'DDR4-2133, LPDDR3-1866', 'DDR4-3200, LPDDR4x-3733',
       'LPDDR4x-4267', 'DDR4 2133, DDR3L 1333/1600 @ 1.35V',
       'DDR4-2133, DDR3L-1600', 'DDR4-3200, LPDDR4x-4267',
       'DDR4-3200, LPDDR4x-4267, In-Band ECC', 'DDR4', 'DDR-2400',
       'Up to 3200 MT/s', 'DDR4-3200, LPDDR4x-3733, In -Band ECC',
       'DDR4-2133', 'DDR4 2933', 'DDR4 2666', 'DDR4 1600/1866/2133/2400',
       'DDR4 1600/1866/2133/2400/2666', 'DDR4, DDR3'], dtype=object)
```

In [44]:

```
data2.max_memory_size.unique()
```

Out[44]:

```
array([1.67772160e+07, 3.35544320e+07, 6.71088640e+07, 8.05306368e+08,
       1.61061274e+09, 8.38860800e+06, 1.07374182e+09, 9.21698304e+06,
       4.19430400e+06, 1.34217728e+08, 4.83183821e+09, 1.71756749e+07,
       1.20259084e+09,            nan, 6.44245094e+09, 2.68435456e+08,
       6.73500365e+07, 6.75807232e+07, 5.36870912e+08, 2.14748365e+09,
       4.29496730e+09])
```

In [45]:

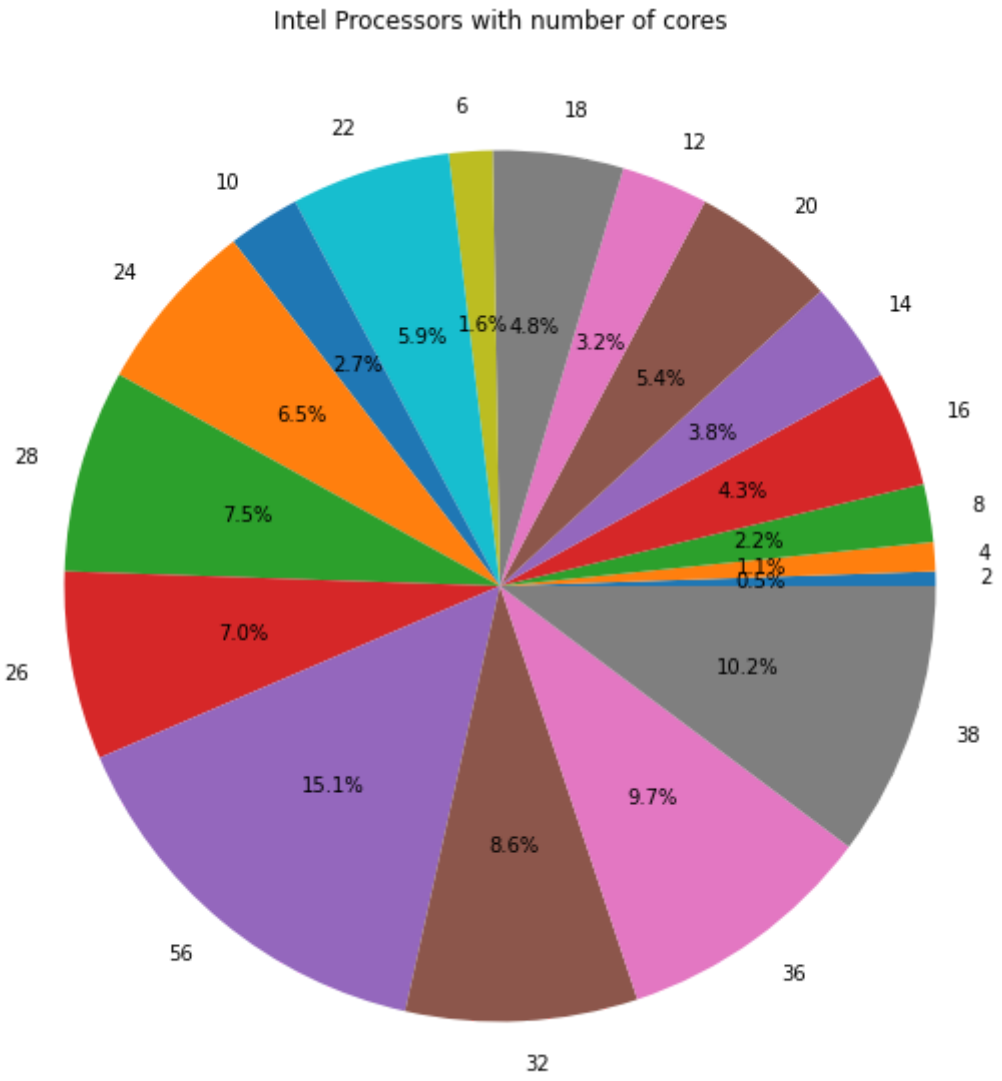
```
data2.drop(['id','url'] ,axis=1 , inplace =True)
```

In [46]:

```
intel_cores = list(data2['cores'].dropna().unique())
```

In [47]:

```
plt.figure(figsize =(15, 10))
plt.pie(intel_cores, labels = intel_cores , autopct='%1.1f%%')
plt.title('Intel Processors with number of cores')
plt.show()
```



In [48]:

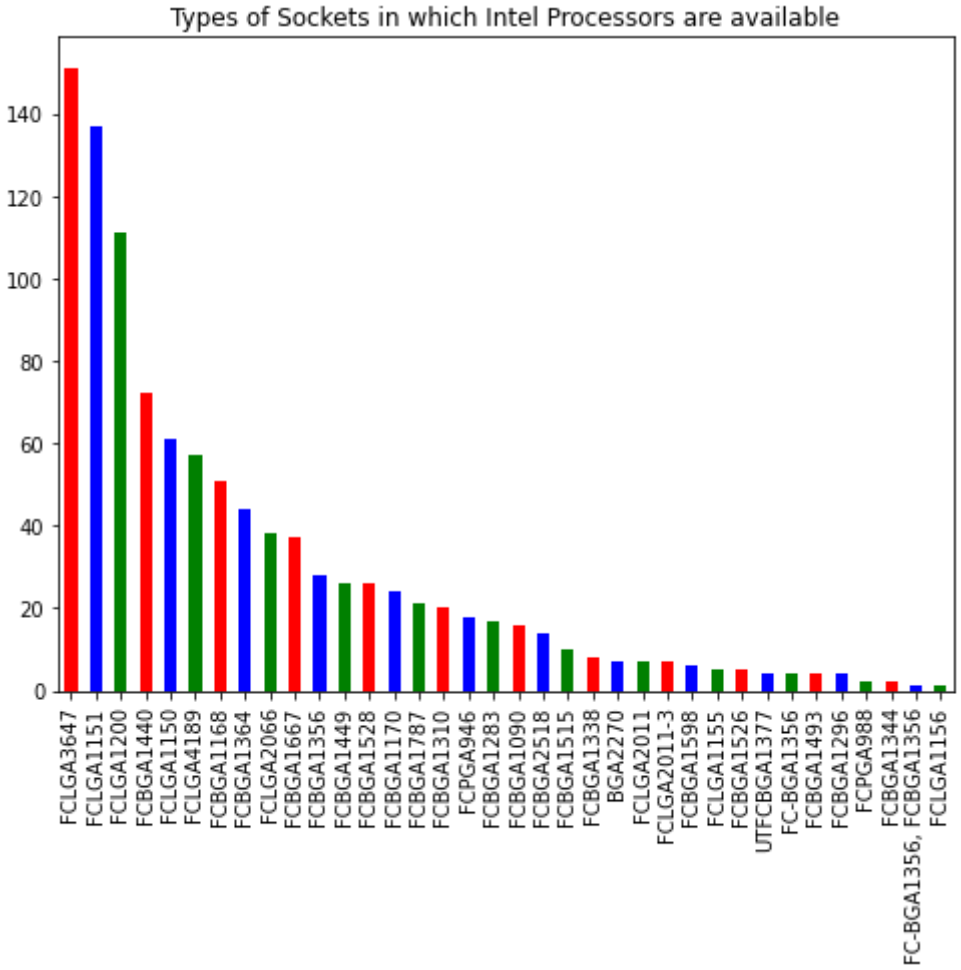
```
text = " ".join(review for review in data2.name.astype(str))
stopwords = set(STOPWORDS)
```

In [49]:

```
wordcloud = WordCloud(stopwords=stopwords, background_color="white", width=800, height=4
plt.axis("off")
plt.tight_layout(pad=0)
plt.imshow(wordcloud, interpolation='bilinear')
plt.show()
```

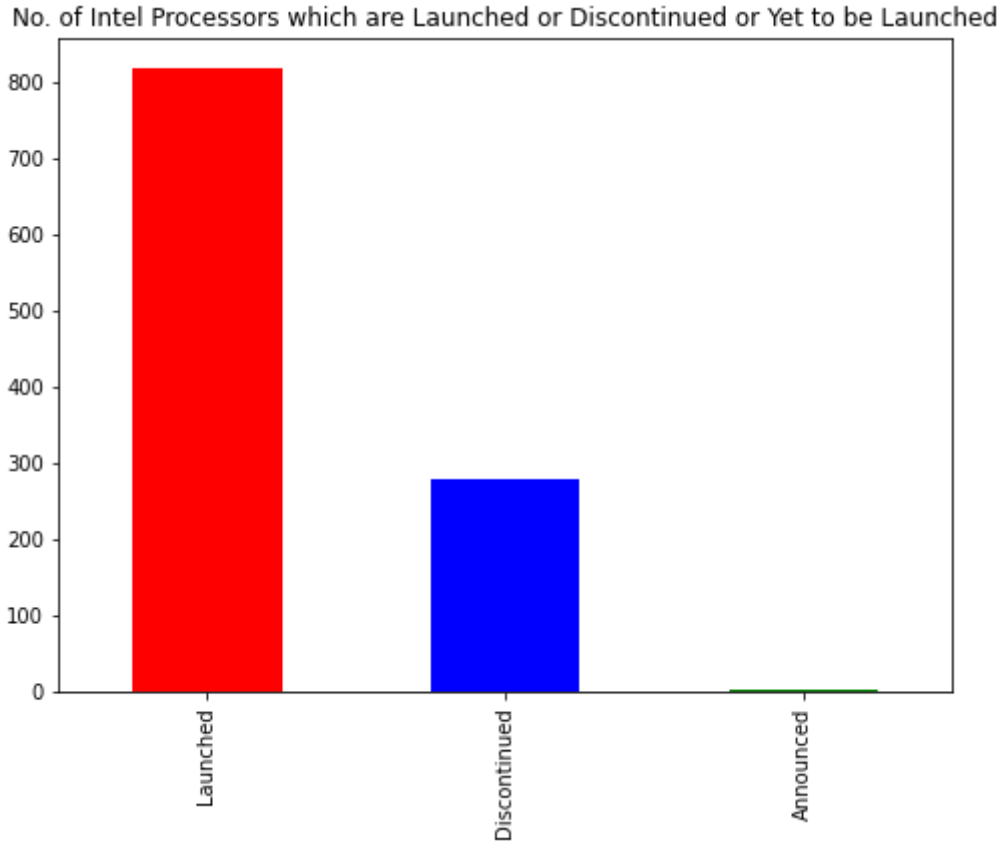
In [50]:

```
plt.figure(figsize =(8, 6))
data2['socket'].value_counts().plot.bar(color=['red','blue','green'])
plt.title('Types of Sockets in which Intel Processors are available')
plt.show()
```



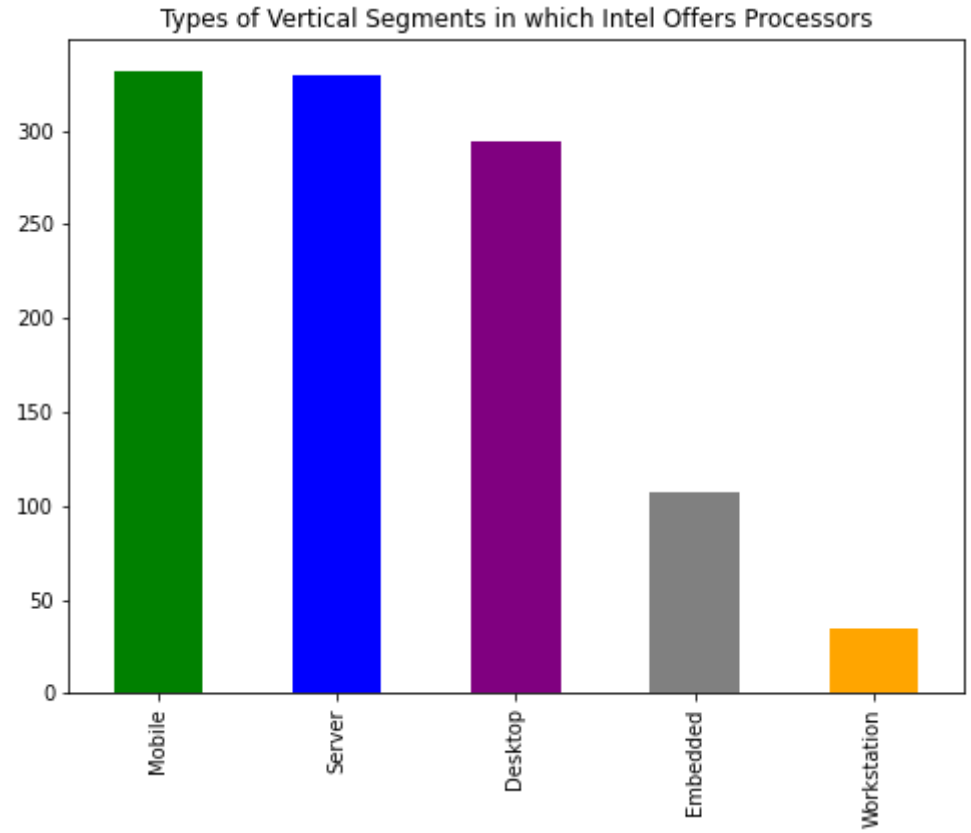
In [52]:

```
plt.figure(figsize =(8, 6))
data2['status'].value_counts().plot.bar(color=['red','blue','green'])
plt.title('No. of Intel Processors which are Launched or Discontinued or Yet to be Launc
plt.show()
```



In [53]:

```
plt.figure(figsize =(8, 6))
data2['vertical_segment'].value_counts().plot.bar(color=['green','blue','purple','gray',
plt.title('Types of Vertical Segments in which Intel Offers Processors')
plt.show()
```



In [54]:

```
sns.relplot(
    data=data2,
    x="launch_date", y="sku",
    kind="line", size_order=["T1", "T2"],
    height=5, aspect=1.3, facet_kws=dict(sharex=False),
)
plt.xticks(rotation = 90)
plt.title('SKU Benchamrks over the Years for Intel Processors')
plt.show()
```