Final Project: Semiconductor Company Sales Analysis by Kaelan Yim

```
for COMP 122 - Spring 2023
```

Table of Contents

Introduction

I will be using a chart of semiconductor company market revenue worldwide from 2009 to 2022 from Statista. Some companies I expect to see on there are well-known chip manufacturers like Intel, Samsung, and Micron.

The questions I will analyze are:

1. Which company grew the most from 2009 to 2022?
2. How this number compares to the average of the company's year-over-year gains?
3. Which company produced the most revenue cumulatively?

Import Required Libraries: The first step of coding is to import the required libraries such as Pandas, NumPy, and Matplotlib. These libraries provide useful functions for data analysis and visualization.

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

Load the Dataset: Use the Pandas library to load the dataset into a Pandas DataFrame object.

In [ ]:
```
df = pd.read_csv(r'.\semicon.csv')
```

Explore the Dataset: Use various Pandas functions to explore the dataset, including:

head() and **tail()** functions to view the first and last few rows of the dataset.

info() function to get information about the data types of each column and the number of non-null values in each column.

describe() function to get a summary of the dataset's statistical measures such as mean, standard deviation, minimum and maximum values, etc.

value_counts() function to get the frequency of each unique value in a particular column.

In [ ]:
```
print(df.info())
print(df.head(5))
print(df.describe())
# tail and value_counts aren't relevant here
# the output is severely truncated, the print for describe doesn't even show up bel
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 19 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Year                    14 non-null     int64  
 1   Samsung Electronics     14 non-null     float64
 2   Intel                   14 non-null     float64
 3   SK Hynix                14 non-null     float64
 4   Qualcomm                14 non-null     float64
 5   Micron Technology       14 non-null     float64
 6   Broadcom                14 non-null     float64
 7   AMD                     3 non-null      float64
 8   Texas Instruments       14 non-null     float64
 9   MediaTek                5 non-null      float64
 10  Apple                   2 non-null      float64
 11  STMicroelectronics      9 non-null      float64
 12  Western Digital         2 non-null      float64
 13  Renesas Technology      6 non-null      float64
 14  Infineon                2 non-null      float64
 15  NXP                     4 non-null      float64
 16  Kioxia (Toshiba Memory)  10 non-null     float64
 17  Nvidia                  3 non-null      float64
 18  Others                  14 non-null     float64
dtypes: float64(18), int64(1)
memory usage: 2.2 KB
None
   Year  Samsung Electronics  Intel  SK Hynix  Qualcomm  Micron Technology  \
0  2009                17.75  33.43      6.04      6.41               4.17   
1  2010                28.10  42.00      9.90      7.20               8.20   
2  2011                27.76  50.67      9.39     10.00               7.64   
3  2012                28.62  49.09      8.97     13.18               6.92   
4  2013                30.64  48.59     12.63     17.21              11.92   

   Broadcom  AMD  Texas Instruments  MediaTek  Apple  STMicroelectronics  \
0      4.32  NaN               9.14       NaN    NaN                8.46   
1      6.60  NaN              11.90       NaN    NaN               10.30   
2      7.16  NaN              11.75       NaN    NaN                9.64   
3      7.85  NaN              11.11       NaN    NaN                8.42   
4      8.20  NaN              10.59       NaN    NaN                8.08   

   Western Digital  Renesas Technology  Infineon  NXP  \
0              NaN                4.54       NaN  NaN   
1              NaN               10.20       NaN  NaN   
2              NaN               10.65       NaN  NaN   
3              NaN                9.15       NaN  NaN   
4              NaN                7.98       NaN  NaN   

   Kioxia (Toshiba Memory)  Nvidia  Others  
0                     9.60     NaN  124.82  
1                    12.40     NaN  152.58  
2                    11.77     NaN  151.34  
3                    10.61     NaN  145.99  
4                    11.28     NaN  148.32  
            Year  Samsung Electronics      Intel   SK Hynix   Qualcomm  \
count    14.0000            14.000000  14.000000  14.000000  14.000000   
```

```
mean   2015.5000            44.913571  55.595000  19.807857  16.384286
std       4.1833            18.579894  11.372546  10.802587   7.376349
min    2009.0000            17.750000  33.430000   6.040000   6.410000
25%    2012.2500            29.125000  49.485000  10.582500  13.287500
50%    2015.5000            38.975000  53.210000  16.180000  15.750000
75%    2018.7500            60.165000  64.400000  26.240000  17.547500
max    2022.0000            73.710000  72.760000  36.350000  34.750000

       Micron Technology   Broadcom        AMD  Texas Instruments   MediaTek
count          14.000000  14.000000   3.000000          14.000000   5.000000  \
mean           16.655714  12.107857  16.420000          12.911429  12.300000
std             8.577648   5.621209   6.810793           2.600275   5.370731
min             4.170000   4.320000   9.670000           9.140000   6.700000
25%             9.130000   7.937500  12.985000          11.532500   7.960000
50%            15.045000  10.825000  16.300000          11.900000  10.990000
75%            22.620000  15.665000  19.795000          13.642500  17.620000
max            29.740000  23.810000  23.290000          18.810000  18.230000

           Apple  STMicroelectronics  Western Digital  Renesas Technology
count   2.000000            9.000000         2.000000            6.000000  \
mean   12.600000            8.520000         6.665000            8.300000
std     7.000357            1.132563         3.528463            2.242115
min     7.650000            6.800000         4.170000            4.540000
25%    10.125000            8.020000         5.417500            7.455000
50%    12.600000            8.420000         6.665000            8.565000
75%    15.075000            9.580000         7.912500            9.937500
max    17.550000           10.300000         9.160000           10.650000

       Infineon       NXP  Kioxia (Toshiba Memory)     Nvidia      Others
count   2.00000  4.000000                10.000000   3.000000   14.000000
mean    6.25000  8.405000                10.361000  11.596667  179.997143
std     0.79196  1.264186                 1.323585   4.816787   46.785746
min     5.69000  6.540000                 7.830000   7.330000  124.820000
25%     5.97000  8.197500                 9.680000   8.985000  149.075000
50%     6.25000  8.885000                10.490000  10.640000  158.825000
75%     6.53000  9.092500                11.127500  13.730000  196.832500
max     6.81000  9.310000                12.400000  16.820000  277.500000
```

Data Wrangling

Clean the Dataset: This step involves cleaning the dataset by handling missing or duplicate values, fixing data types, and removing irrelevant columns.
Some of the common data cleaning techniques include:

Dropping duplicates using **drop_duplicates()** function.
Handling missing values using **fillna()** function or by removing rows with missing values using **dropna()**function.

Converting data types of columns using **astype()** function.

Renaming columns using **rename()** function.

Removing irrelevant columns using **drop()** function.

In [ ]:
```
# no things I want to do here
# I would do fillna but it makes it harder to find the yearly average earnings when
```

Exploratory Data Analysis

Analyze the Dataset: Once the dataset is cleaned, it's time to perform some analysis to gain insights. This can involve creating visualizations using Matplotlib and performing statistical analysis to answer your research questions.

Research Question 1: Which company grew the most from 2009 to 2022?

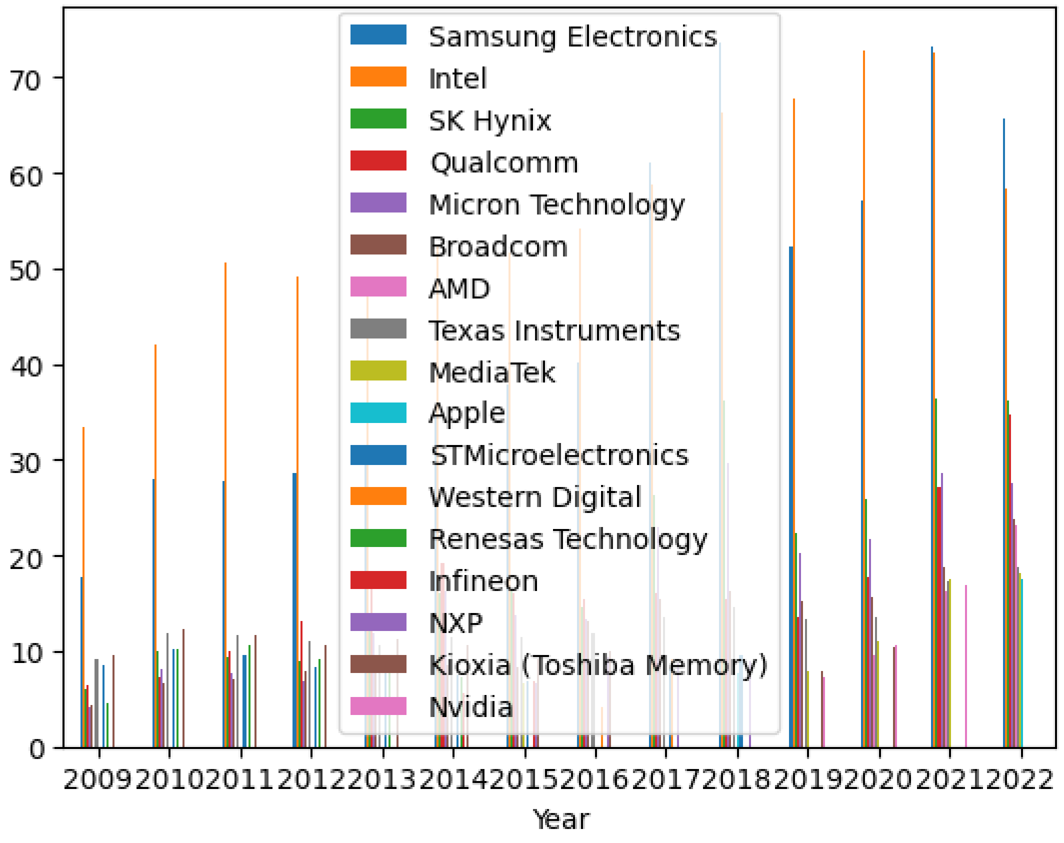
In [ ]:
```
amogus = df.columns.tolist()
amogus = amogus[1:-1]
ax = df.plot.bar(x="Year", y=amogus, rot=0)
# not sure how to reposition the legend

sus1 = 0
sus2 = 0
sus3 = ""
sus4 = ""
sus5 = 0
sus6 = 0

for column in df.columns[1:-1]:
    begval = df[column].iloc[0]
    endval = df[column].iloc[-1]
    sus1 = endval - begval
    sus3 = column
    if sus1 > sus2:
        sus2 = sus1
        sus4 = sus3
        sus5 = begval
        sus6 = endval
# yes I excluded "Year" and "Others" since neither are really applicable here

sus7 = ((sus6 - sus5) / sus5) * 100
print("The company with the largest growth was", sus4, "with", sus2, "billion dolla
print("This was a", sus7, "percent increase in revenue.")
```

```
The company with the largest growth was Samsung Electronics with 47.84 billion dolla
rs more revenue in the final reported year than the first reported year.
This was a 269.5211267605634 percent increase in revenue.
```



Research Question 2: How does this number compares to the average of the company's year-over-year gains?

In [ ]:

```
sus8 = 0
sus9 = 0
sus10 = ""
sus11 = ""

for column in df.columns[1:-1]:
    amo1 = df[column].diff()
    avgchg = amo1.mean()
    sus8 = avgchg
    sus10 = column
    if sus8 > sus9:
        sus9 = sus8
        sus11 = sus10

amo2 = df[sus3].diff()
amo3 = amo2.mean()
sus12 = amo3

print("The company with the highest year-over-year gain was", sus11, "at", sus9, "p
print("The previous company mentioned above (", sus4, ") had a", sus12, "annual rev
```

```
The company with the highest year-over-year gain was AMD at 6.81 percent per year.
The previous company mentioned above ( Samsung Electronics ) had a 4.745 annual reve
nue increase.
```

Research Question 3: Which company produced the most revenue cumulatively?

```
In [ ]: sus13 = 0
        sus14 = ""

        for column in df.columns[1:-1]:
            coltot = df[column].sum()
            if coltot > sus13:
                sus13 = coltot
                sus14 = column

        print(sus14, "had the highest total cumulative revenue at", sus13, "billion dollars
```

```
Intel had the highest total cumulative revenue at 778.33 billion dollars over the re
ported time period.
```

Conclusions

Draw Conclusion: Finally, summarize your findings and draw conclusions based on your analysis.

If I had to invest in only one of the companies on the datasheet as a consumer, I would likely invest in **Samsung** because of the large growth the company has experienced. **AMD** would also be a good choice from the data results above but because it only has entries for three years of data the numbers could be unreliable. If I were a hedge fund manager, I would invest in **Intel** because they have produced earnings reliably for a long time.