HW02 [ECE 720]

Digvijay Anand
200478940

Q1. Add statistics to results.csv

Step 1: Modify Makefile to add an entry into results.csv for layer met1 wirelength
Makefile:

```
…
setup:
	echo  …,route_opt__area__wirelength__tot,route_opt__area__met1__wirelength__tot >
results.csv
…
```

Step 2: Define a python function to remove any garbage entry due to failed runs
clean_results_csv.py:

```
import pandas as pd
import sys

# take argument for filename.csv to clean
csv_filename = sys.argv[1]
df = pd.read_csv(csv_filename)

# remove rows with NaN
df_cleaned = df.dropna()

# create a cleaned copy of results.csv
output_file = csv_filename
df_cleaned.to_csv(output_file, index=False)
```

Step 3: Create target in Makefile to run ‘clean_results_csv.py’ after pnr
Makefile:

```
…
clean_csv: report
	@echo ""
```

```
        @echo "####################"
        @echo ""
        python3 clean_results_csv.py results.csv
        @echo ""
        @echo "Cleaned results.csv"
        @echo ""
        @echo "####################"
…
```

Step 4: Modify parse_reports.py to gather the layer met1 wirelength

parse_reports.py:

```
…
f=open('icc2rm/rpts_icc2/route_opt.check_routes')
for line in f:
        m=re.search(r'TOTAL VIOLATIONS =\s+(\d+)',line)
        if m:
                viol=m.group(1)
                continue
        m=re.search(r'Total Routed Wire Length =\s+(\d+)',line)
        if m:
                wire_len=m.group(1)
                continue

        m=re.search(r'\s*Layer\s*met1\s*:\s*(\d+)',line)
        if m:
                met1_wire_len=m.group(1)
                print('########################################')
                print('Q1:')
                print(f'route_opt__area__met1__wirelength__tot = {met1_wire_len}')
                print('########################################')
                break
f.close()
…
results.write(met1_wire_len+'\n')                # route_opt__area__met1__wirelength__tot
…
```

Result:

```
python3 parse_reports.py 25 0.55 met5 0.5 0.1
########################################
Q1:
route_opt__area__met1__wirelength__tot = 91386
########################################
```

```
######################
python3 clean_results_csv.py results.csv
Cleaned results.csv
######################
```

Q2. New design and statistics for xbar:

Step 1: make changes to other files based on instruction in pdrm/README.md for modifying the flow
Step 2: default settings: CLK_PER = 25, MAXLYR = met5, CLKUNCERT = 0.1
Step 3: change UTIL argument while running make to get the highest core utilization
Step 4: check if DRC == 0

Bash run:

```
make UTIL=0.55
```

Result:

```
UTIL (max)                                          = 0.55
Final Utilization (from route_opt.report_utilization)  = 0.6127
```

Note:

- Although it ran once for UTIL=0.6 and Final utilization of 0.665, the result was not consistent as another similar gave non-zero DRC violation.
- Similarly, for CLK_PER == 10 / 5, the max utilization was stable forUTIL = 0.55

Q3. Plot histogram and find average wire length:

Step 1: Download Rocket.def to /hw02/p3/
Step 2: source setup.sh
Step 3: Create a histogram for different wirelength [figure 1]
listshapes.py:

```
total = 0
x_nn = []
length_per_nn = []

for nn in netlength:
  if nn not in ('VDD','VSS'):
    length_per_nn.append(netlength[nn])
    x_nn.append(nn)
    total+=netlength[nn]
df = pd.DataFrame(list(netlength.items()), columns=['NetName', 'NetLength'])
df_cleaned = df[~df['NetName'].isin(['VDD', 'VSS'])]

########################################
# define and save plot
########################################
ax = df_cleaned['NetLength'].plot(kind='hist', bins=128, color='blue', edgecolor='black', figsize=(12,
9))
ax.set_xlim(0, 128)
ax.set_xlabel('Net Length', fontweight='bold', fontsize=18)
ax.set_ylabel('Frequency', fontweight='bold', fontsize=18)
ax.set_title('Histogram of Net Length', fontweight='bold', fontsize=24)
ax.tick_params(axis='x', labelsize=14)
ax.tick_params(axis='y', labelsize=14)
for label in ax.get_xticklabels():
    label.set_fontweight('bold')
for label in ax.get_yticklabels():
    label.set_fontweight('bold')
filename = f'netlength_histogram_pandas.png'

plt.savefig(filename, dpi=300)
```

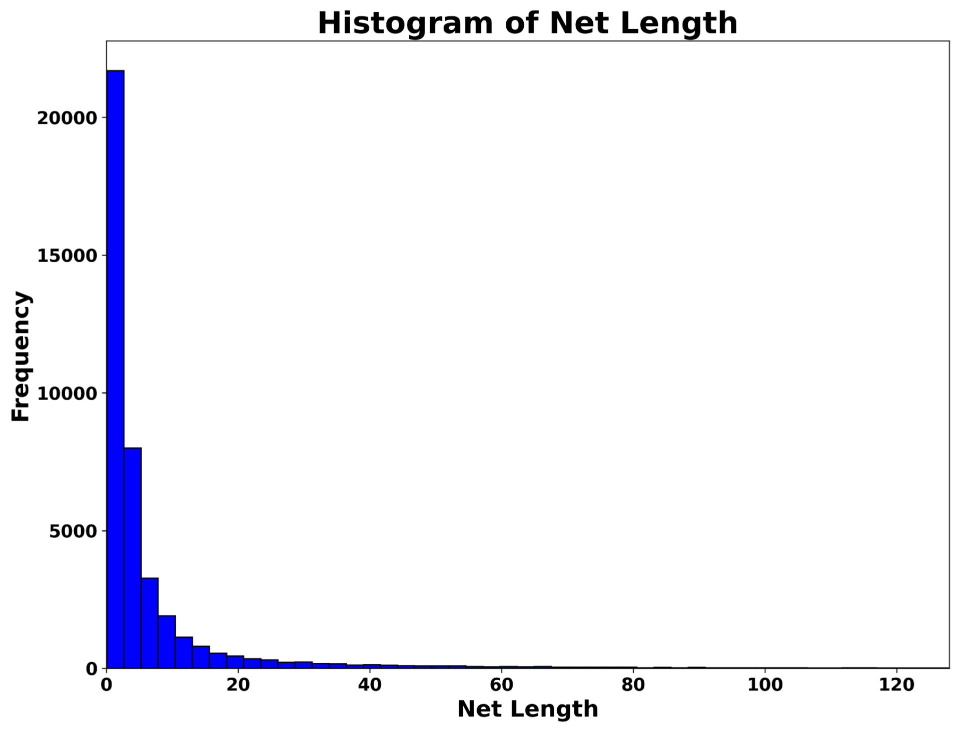

Figure 1. Histogram of Net Length

Step 4: Calculate average wirelength:
listshapes.py:

```
…
num_of_total_cells = 40197
total_per_cell = total/num_of_total_cells
…
```

Step 5: Get average wire length using Donath’s method:
listshapes.py:

```
########################################
# define Donath method:
########################################

def L_avg(C, p, d_avg):
   if p != 1/2:
      term1 = 7 * (C**(p-1/2) - 1) / (4**(p-1/2) - 1)
      term2 = (1 - C**(p-3/2)) / (1 - 4**(p-3/2))
      term3 = (1 - 4**(p-1)) / (1 - C**(p-1))
      L_avg = d_avg * (2/9) * (term1 - term2) * term3

   else:
      term1 = 7 * np.log2(C) / 2  # log4(C) = log2(C) / 2
      term2 = (1 - C**(p-3/2)) / (1 - 4**(p-3/2))
      term3 = (1 - 4**(p-1)) / (1 - C**(p-1))
      L_avg = d_avg * (2/9) * (term1 - term2) * term3
   return L_avg

########################################
# use given values for d_avg:
########################################

corearea = 152.64*152.064
C = 40197
d_avg = np.sqrt(corearea/C)
p1 = 0.5
p2 = 0.75

result1 = L_avg(C, p1, d_avg)
result2 = L_avg(C, p2, d_avg)
```

Result Q3:

based on Listshape.py:

Total length	= 313515.8450 microns	[Total Wire Length]
L_avg	= 7.7995 microns	[Average Wire Length Per Cell]

Donath's method for wirelength:

L_avg (with p = 1/2)	= 4.4293 microns
L_avg (with p = 3/4)	= 11.7527 microns