Mini Project Report:

Kernighan-Lin (KL) Algorithm in C++

Introduction

The Kernighan-Lin (KL) algorithm performs partitioning through iterative improvement steps. It was proposed by B. W. Kernighan and S. Lin in 1970 for bi-partitioning (k = 2) graphs.
It is a well-known algorithm used by EDA tools for partitioning. It is implemented in this project.

Implementation:

- The program is generic and is not specific to some kind or number of netlist, it can do as many passes as defined in the algorithm.
- The program can handle the odd number of components in a netlist by adding dummy gate prior initial partitioning

Results

1)
```
(a: IN1 N2, IN2 N3, OUT N4)
(b: IN1 N2, IN2 N1, OUT N3)
(c: IN1 N5, IN2 N6, OUT N7)
(d: IN1 N4, IN2 N5, OUT N6)
```

<u>The result of applying the algorithm on the given netlist above:</u>

```
Number of Gates: 4

Gate 1 Name: a          IN1:    2       IN2:    3       OUT:    4
Gate 2 Name: b          IN1:    2       IN2:    1       OUT:    3
Gate 3 Name: c          IN1:    5       IN2:    6       OUT:    7
Gate 4 Name: d          IN1:    4       IN2:    5       OUT:    6

Connectivity Matrix:
        a       b       c       d
a       0       2       0       1
b       2       0       0       0
c       0       0       0       2
d       1       0       2       0

||================================ Applying Kernighan-Lin (KL) Algorithm ================================||

Initial Partitions:
Partition A:
Gate 1 Name: a
Gate 2 Name: b

Partition B:
Gate 1 Name: c
Gate 2 Name: d

|-------------------- Pass 1 ---------------------|

*********************************************
Iteration: 0     Unfixed gates:  c  a  d  b
*********************************************
Cut nets of a = 1
Cut nets of b = 0
Cut nets of c = 0
Cut nets of d = 1
Uncut nets of a = 2
Uncut nets of b = 2
Uncut nets of c = 2
Uncut nets of d = 2
```

```
Cut cost & delta_g :
D(a) = -1,      D(c) = -2
D(b) = -2,      D(d) = -1
delta_g(a,c) = -3
delta_g(a,d) = -4
delta_g(b,c) = -4
delta_g(b,d) = -3

Highest gain is: -3  --  Pair to switch are: a and c
Swap and Fix a and c
G = -3

*********************************************
Iteration: 1     Unfixed gates:  d  b
*********************************************
Cut nets of b = 2
Cut nets of d = 2
Uncut nets of b = 0
Uncut nets of d = 1

Cut cost & delta_g :
D(b) = 2,       D(d) = 1
delta_g(b,d) = 3

Highest gain is: 3  --  Pair to switch are: b and d
Swap and Fix b and d
G = 0
|------------------ End of Pass 1 ------------------|

G_m = 0

Configuration after Pass 1 :
Partition A:
Gate 1 Name: c
Gate 2 Name: d

Partition B:
Gate 1 Name: a
Gate 2 Name: b


****************** Best configuration after 1 Pass(es) *******************
Partition A:
Gate 1 Name: c
Gate 2 Name: d

Partition B:
Gate 1 Name: a
Gate 2 Name: b
```

- As shown in the result:
  The alg. Performs 1 pass only to solve for the given netlist

```
2)    (a: IN1 N2, IN2 N3, OUT N4)
      (c: IN1 N5, IN2 N6, OUT N7)
      (b: IN1 N2, IN2 N1, OUT N3)
      (d: IN1 N4, IN2 N5, OUT N6)
```

<u>The result of applying the algorithm on the given netlist above (swaped c and b):</u>

```
Number of Lines: 4
Number of Pins:  0
Number of Gates: 4

Gate 1 Name: a        IN1:    2      IN2:    3      OUT:    4
Gate 2 Name: c        IN1:    5      IN2:    6      OUT:    7
Gate 3 Name: b        IN1:    2      IN2:    1      OUT:    3
Gate 4 Name: d        IN1:    4      IN2:    5      OUT:    6

Connectivity Matrix:
        a       c       b       d
a       0       0       2       1
c       0       0       0       2
b       2       0       0       0
d       1       2       0       0

||================================ Applying Kernighan-Lin (KL) Algorithm ================================||

Initial Partitions:
Partition A:
Gate 1 Name: a
Gate 2 Name: c

Partition B:
Gate 1 Name: b
Gate 2 Name: d

|-------------------- Pass 1 ---------------------|

********************************************
Iteration: 0     Unfixed gates:  b  a  d  c
********************************************
Cut nets of a = 3
Cut nets of c = 2
Cut nets of b = 2
Cut nets of d = 3
Uncut nets of a = 0
Uncut nets of c = 0
Uncut nets of b = 0
Uncut nets of d = 0

Cut cost & delta_g :
D(a) = 3,       D(b) = 2
D(c) = 2,       D(d) = 3
delta_g(a,b) = 1
delta_g(a,d) = 4
delta_g(c,b) = 4
delta_g(c,d) = 1


Highest gain is: 4  --  Pair to switch are: a and d
Swap and Fix a and d
G = 4
```

```
***********************************************
Iteration: 1     Unfixed gates:  b  c
***********************************************
Cut nets of c = 0
Cut nets of b = 0
Uncut nets of c = 2
Uncut nets of b = 2

Cut cost & delta_g :
D(b) = -2
D(c) = -2,      delta_g(c,b) = -4

Highest gain is: -4  --  Pair to switch are: c and b
Swap and Fix c and b
G = 0
|----------------- End of Pass 1 -----------------|

G_m = 4

Configuration after Pass 1 :
Partition A:
Gate 1 Name: d
Gate 2 Name: c

Partition B:
Gate 1 Name: b
Gate 2 Name: a

|-------------------- Pass 2 ---------------------|

***********************************************
Iteration: 0     Unfixed gates:  b  d  a  c
***********************************************
Cut nets of d = 1
Cut nets of c = 0
Cut nets of b = 0
Cut nets of a = 1
Uncut nets of d = 2
Uncut nets of c = 2
Uncut nets of b = 2
Uncut nets of a = 2

Cut cost & delta_g :
D(d) = -1,      D(b) = -2
D(c) = -2,      D(a) = -1
delta_g(d,b) = -3
delta_g(d,a) = -4
delta_g(c,b) = -4
delta_g(c,a) = -3

Highest gain is: -3  --  Pair to switch are: d and b
Swap and Fix d and b
G = -3

***********************************************
Iteration: 1     Unfixed gates:  a  c
***********************************************
Cut nets of c = 2
Cut nets of a = 2
Uncut nets of c = 0
Uncut nets of a = 1

Cut cost & delta_g :
D(c) = 2,       D(a) = 1
delta_g(c,a) = 3

Highest gain is: 3  --  Pair to switch are: c and a
Swap and Fix c and a
G = 0
|----------------- End of Pass 2 -----------------|
```

```
G_m = 0

Configuration after Pass 2 :
Partition A:
Gate 1 Name: b
Gate 2 Name: a

Partition B:
Gate 1 Name: d
Gate 2 Name: c


****************** Best configuration after 2 Pass(es) *******************
Partition A:
Gate 1 Name: b
Gate 2 Name: a

Partition B:
Gate 1 Name: d
Gate 2 Name: c
```

- As shown in the result:
  The alg. Performs 2 pass to solve for the rearranged netlist

3)
```
(a: IN1 N2, IN2 N3, OUT N4)
(b: IN1 N2, IN2 N1, OUT N3)
(d: IN1 N4, IN2 N5, OUT N6)
```

<u>The result of applying the algorithm on the given netlist above (deleted c to make odd number):</u>

```
Number of Lines: 3
Number of Pins:  0
Number of Gates: 3

Gate 1 Name: a          IN1:    2       IN2:    3       OUT:    4
Gate 2 Name: b          IN1:    2       IN2:    1       OUT:    3
Gate 3 Name: d          IN1:    4       IN2:    5       OUT:    6

Connectivity Matrix:
        a       b       d
a       0       2       1
b       2       0       0
d       1       0       0

||================================= Applying Kernighan-Lin (KL) Algorithm =================================||

Initial Partitions:
Partition A:
Gate 1 Name: a
Gate 2 Name: b

Partition B:
Gate 1 Name: d
Gate 2 Name: dummy
```

```
|-------------------- Pass 1 ---------------------|

********************************************
Iteration: 0     Unfixed gates:  d  a  dummy  b
********************************************
Cut nets of a = 1
Cut nets of b = 0
Cut nets of d = 1
Cut nets of dummy = 0
Uncut nets of a = 2
Uncut nets of b = 2
Uncut nets of d = 0
Uncut nets of dummy = 0

Cut cost & delta_g :
D(a) = -1,       D(d) = 1
D(b) = -2,       D(dummy) = 0
delta_g(a,d) = -2
delta_g(a,dummy) = -1
delta_g(b,d) = -1
delta_g(b,dummy) = -2

Highest gain is: -1  --  Pair to switch are: a and dummy
Swap and Fix a and dummy
G = -1

********************************************
Iteration: 1     Unfixed gates:  d  b
********************************************
Cut nets of b = 2
Cut nets of d = 0
Uncut nets of b = 0
Uncut nets of d = 1

Cut cost & delta_g :
D(d) = -1
D(b) = 2,        delta_g(b,d) = 1

Highest gain is: 1  --  Pair to switch are: b and d
Swap and Fix b and d
G = 0
|----------------- End of Pass 1 -----------------|

G_m = 0

Configuration after Pass 1 :
Partition A:
Gate 1 Name: dummy
Gate 2 Name: d

Partition B:
Gate 1 Name: b
Gate 2 Name: a


****************** Best configuration after 1 Pass(es) *******************
Partition A:
Gate 1 Name: dummy
Gate 2 Name: d

Partition B:
Gate 1 Name: b
Gate 2 Name: a
```

- As shown in the result:
  The alg. Addes a dummy gate and performs 1 pass to solve for the netlist