ASIC CONSTRUCTION

15

Key terms and concepts:

- A microelectronic system (or system on a chip) is the town and ASICs (or system blocks) are the buildings
- **System partitioning** corresponds to town planning.
- **Floorplanning** is the architect's job.
- **Placement** is done by the builder.
- **Routing** is done by the electrician.

15.1 Physical Design

Key terms and concepts: Divide and conquer • system partitioning • floorplanning • chip planning • placement • routing • global routing • detailed routing

15.2 CAD Tools

Key terms and concepts: **goals** and **objectives** for each physical design step

System partitioning:

- Goal. Partition a system into a number of ASICs.
- Objectives. Minimize the number of external connections between the ASICs. Keep each ASIC smaller than a maximum size.

Floorplanning:

- Goal. Calculate the sizes of all the blocks and assign them locations.
- Objective. Keep the highly connected blocks physically close to each other.

Placement:

- Goal. Assign the interconnect areas and the location of all the logic cells within the flexible blocks.
- Objectives. Minimize the ASIC area and the interconnect density.

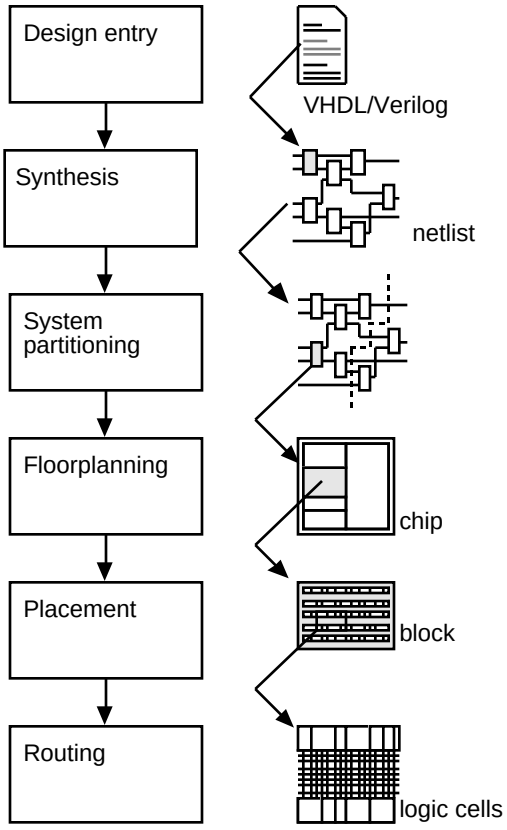
Part of an ASIC design flow showing the system partitioning, floorplanning, placement, and routing steps.

These steps may be performed in a slightly different order, iterated or omitted depending on the type and size of the system and its ASICs.

As the focus shifts from logic to interconnect, floorplanning assumes an increasingly important role.

Each of the steps shown in the figure must be performed and each depends on the previous step.

However, the trend is toward completing these steps in a parallel fashion and iterating, rather than in a sequential manner.



Global routing:

- Goal. Determine the location of all the interconnect.
- Objective. Minimize the total interconnect area used.

Detailed routing:

- Goal. Completely route all the interconnect on the chip.
- Objective. Minimize the total interconnect length used.

15.2.1 Methods and Algorithms

Key terms and concepts: **methods** or **algorithms** are exact or heuristic (algorithm is usually reserved for a method that always gives a solution)• The complexity O(f(n)) is important because *n* is very large • algorithms may be constant, logarithmic, linear, or quadratic in time• many VLSI problems are **NP-complete** • we need **metrics**: a **measurement function** or objective function, a **cost function** or gain function, and possibly **constraints**

15.3 System Partitioning

Key terms and concepts: **partitioning** • we can't do "What is the cheapest way to build my system?" • we can do "How do I split this circuit into pieces that will fit on a chip?"

System partitioning for the Sun Microsystems SPARCstation 1

	SPARCstation 1 ASIC	Gates /k-gate	Pins	Package	Type
1	SPARC IU (integer unit)	20	179	PGA	CBIC
2	SPARC FPU (floating-point unit)	50	144	PGA	FC
3	Cache controller	9	160	PQFP	GA
4	MMU (memory-management unit)	5	120	PQFP	GA
5	Data buffer	3	120	PQFP	GA
6	DMA (direct memory access) controller	9	120	PQFP	GA
7	Video controller/data buffer	4	120	PQFP	GA
8	RAM controller	1	100	PQFP	GA
9	Clock generator	1	44	PLCC	GA

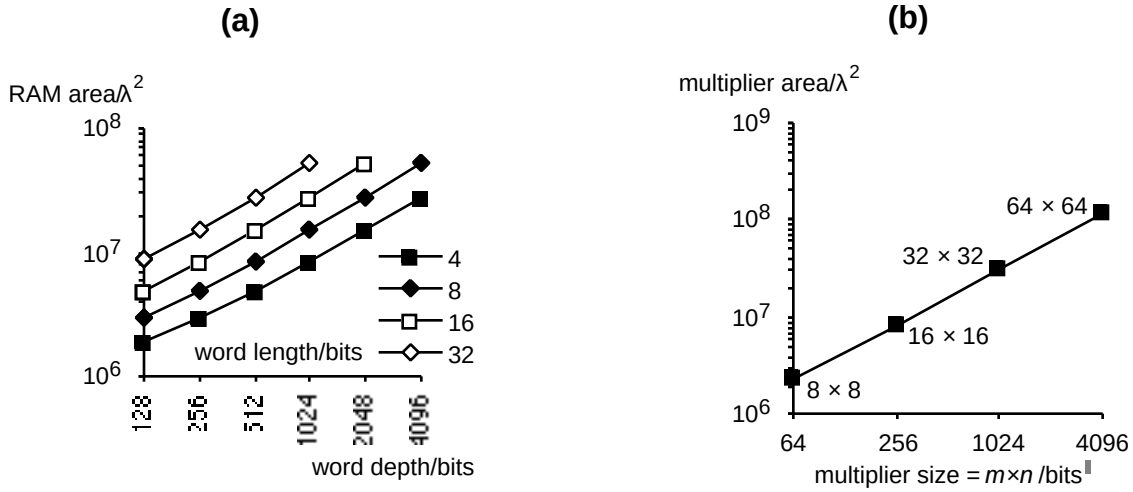
15.4 Estimating ASIC Size

System partitioning for the Sun Microsystems SPARCstation 10

	SPARCstation 10 ASIC	Gates	Pins	Package	Type
1	SuperSPARC Superscalar SPARC	3M-transistors	293	PGA	FC
2	SuperCache cache controller	2M-transistors	369	PGA	FC
3	EMC memory control	40k-gate	299	PGA	GA
4	MSI MBus–SBus interface	40k-gate	223	PGA	GA
5	DMA2 Ethernet, SCSI, parallel port	30k-gate	160	PQFP	GA
6	SEC SBus to 8-bit bus	20k-gate	160	PQFP	GA
7	DBRI dual ISDN interface	72k-gate	132	PQFP	GA
8	MMCodec stereo codec	32k-gate	44	PLCC	FC

Some useful numbers for ASIC estimates, normalized to a 1 µm technology

Parameter	Typical value	Comment	Scaling
Lambda, λ	0.5 µm=0.5 (minimum feature size)	In a 1µm technology, λ≈ 0.5 µm.	NA
Effective gate length	0.25 to 1.0µm	Less than drawn gate length, usually by about 10 percent.	λ
I/O-pad width (pitch)	5 to 10mil =125 to 250µm	For a 1µm technology, 2LM (λ=0.5 µm). Scales less than linearly with λ.	λ
I/O-pad height	15 to 20mil =375 to 500µm	For a 1µm technology, 2LM (λ=0.5µm). Scales approximately linearly with λ.	λ
Large die	1000 mil/side, 10^6 mil^2	Approximately constant	1
Small die	100 mil/side, 10^4 mil^2	Approximately constant	1
Standard-cell density	1.5×10^{-3} gate/µm^2 =1.0gate/mil^2	For 1µm, 2LM, library $= 4\times10^{-4}$ gate/λ^2 (independent of scaling).	$1/\lambda^2$
Standard-cell density	8×10^{-3} gate/µm^2 = 5.0gate/mil^2	For 0.5 µm, 3LM, library $= 5\times10^{-4}$ gate/λ^2 (independent of scaling).	$1/\lambda^2$
Gate-array utilization	60 to 80% 80 to 90%	For 2LM, approximately constant For 3LM, approximately constant	1 1
Gate-array density	(0.8 to 0.9) × standard cell density	For the same process as standard cells	1
Standard-cell routing factor=(cell area+route area)/cell area	1.5 to 2.5 (2LM) 1.0 to 2.0 (3LM)	Approximately constant	1
Package cost	$0.01/pin, “penny per pin”	Varies widely, figure is for low-cost plastic package, approximately constant	1
Wafer cost	$1k to $5k average $2k	Varies widely, figure is for a mature, 2LM CMOS process, approximately constant	1



Estmating circuit size

(a) ASIC memory size. These figures are for static RAM constructed using compilers in a 2LM ASIC process, but with no special memory design rules.

The actual area of a RAM will depend on the speed and number of read–write ports.

(b) Multiplier size for a 2LM process.

The actual area will depend on the multiplier architecture and speed.

15.5 Power Dissipation

Key terms and concepts: dynamic (switching current and short-circuit current) and static (leakage current and subthreshold current) power dissipation

15.5.1 Switching Current

Key terms and concepts: $I = C(\mathrm{d}V/\mathrm{d}t)$ • power dissipation = $0.5\,CV_{DD}^2 = IV = CV(\mathrm{d}V/\mathrm{d}t)$ for one-half the period of the input, $t=1/(2f)$ • total power = $P_1 = fCV^2_{DD}$ • estimate power by counting nodes that toggle

15.5.2 Short-Circuit Current

Key terms and concepts: $P_2 = (1/12)\beta f t_{rf}(V_{DD} - 2\,V_{tn})$ • short-circuit current is typically less than 20 percent of the switching current

15.5.3 Subthreshold and Leakage Current

Key terms and concepts: **subthreshold current** is normally less than 5pAμm^{-1} of gate width • subthreshold current for 10 million transistors (each 10µm wide) is 0.1mA • subthreshold current does not scale • it takes about 120mV to reduce subthreshold current by a factor of 10 • if V_t = 0.36V, at V_{GS}=0 V we can only reduce I_{DS} to 0.001 times its value at V_{GS}=V_t • leakage current • field transistors • quiescent leakage current, I_{DDQ} • **IDDQ test**

15.6 FPGA Partitioning

15.6.1 ATM Simulator

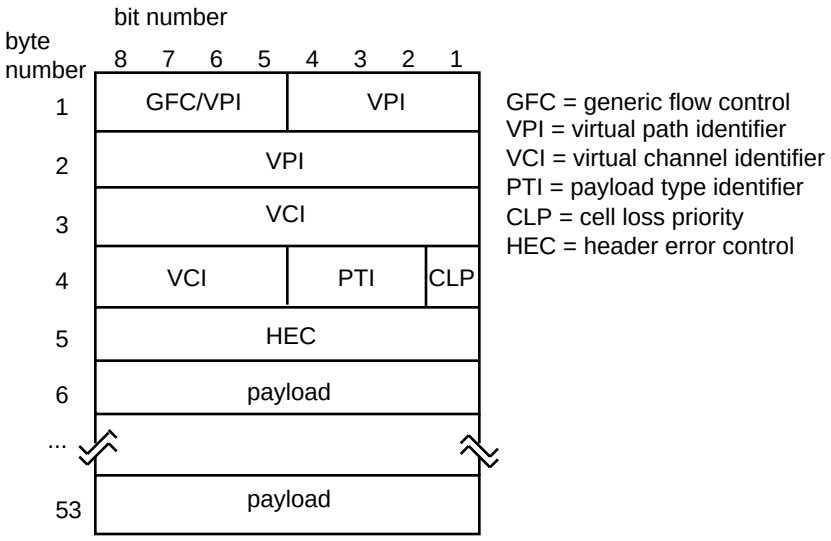
Partitioning of the ATM board using Lattice Logic ispLSI 1048 FPGAs. Each FPGA contains 48 generic logic blocks (GLBs)

Chip #	Size	Chip #	Size
1	42 GLBs	7	36 GLBs
2	64k-bit ×8 SRAM	8	22 GLBs
3	38 GLBs	9	256k-bit × 16 SRAM
4	38 GLBs	10	43 GLBs
5	42 GLBs	11	40 GLBs
6	64k-bit ×16 SRAM	12	30 GLBs

15.6.2 Automatic Partitioning with FPGAs

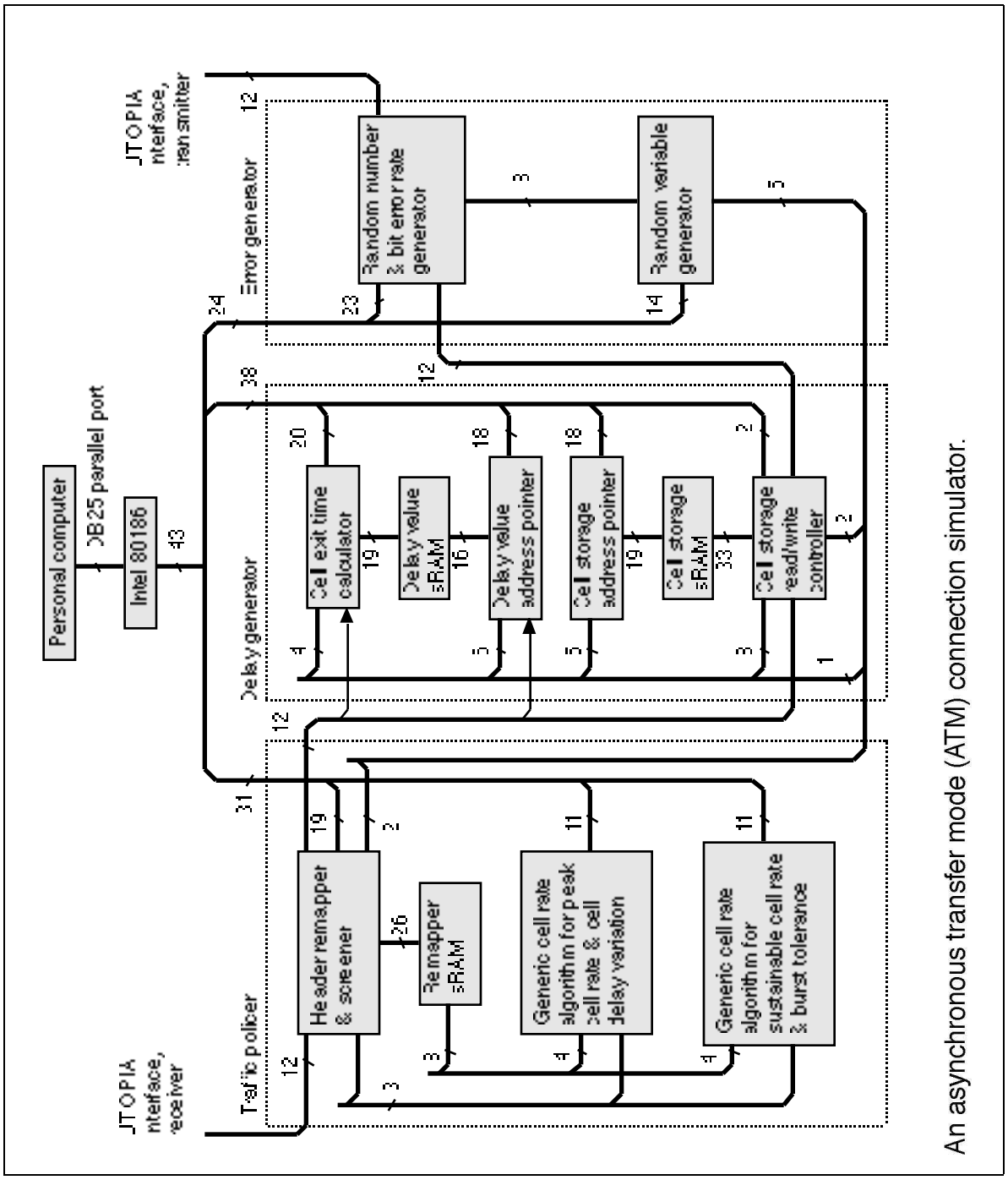
Key terms and concepts: In Altera AHDL you can direct the partitioner to automatically partition logic into chips within the same family, using the AUTO keyword:

```
DEVICE top_level IS AUTO; % let the partitioner assign logic
```



The asynchronous transfer mode (ATM) cell format.

The ATM protocol uses 53-byte cells or packets of information with a data payload and header information for routing and error control.



An asynchronous transfer mode (ATM) connection simulator.

15.7 Partitioning Methods

Key terms and concepts: Examples of goals: A maximum size for each ASIC • A maximum number of ASICs • A maximum number of connections for each ASIC • A maximum number of total connections between all ASICs

15.7.1 Measuring Connectivity

Key terms and concepts: a network has circuit modules (logic cells) and terminals (connectors or pins) • modelled by a graph with vertexes (logic cells) connected by edges (electrical connections, nets or signals) • cutset • net cutset • edge cutset (for the graph) • external connections • internal connections • net cuts • edge cuts

15.7.2 A Simple Partitioning Example

Key terms and concepts: two types of **network partitioning**: **constructive partitioning** and **iterative partitioning improvement**

15.7.3 Constructive Partitioning

Key terms and concepts: **seed growth** or **cluster growth** uses a **seed cell** and forms **clusters** or **cliques** • a useful starting point

15.7.4 Iterative Partitioning Improvement

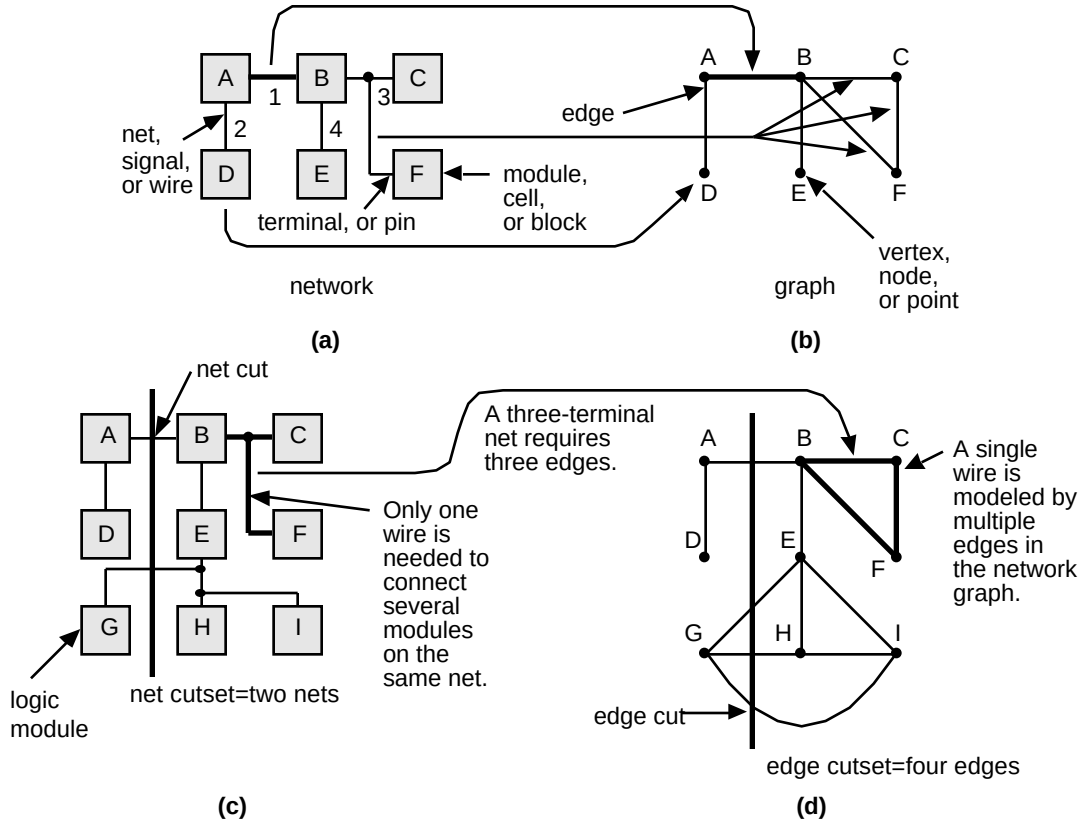
Key terms and concepts: interchange (swap two) and group (swap many) migration • greedy algorithms find a local minimum • group migration algorithms such as the Kernighan–Lin algorithm (basis of min-cut methods) can do better

15.7.5 The Kernighan–Lin Algorithm

Key terms and concepts: a cost matrix plus connectivity matrix models system • measure is the cut cost, or cut weight • careful to distinguish external edge cost and internal edge cost • net-cut partitioning and edge-cut partitioning • hypergraphs with stars, and hyperedges model connections better than edges • the Fiduccia–Mattheyses algorithm uses linked lists to reduce O(K–L algorithm) and is very widely used • base logic cell • balance • critical net

15.7.6 The Ratio-Cut Algorithm

Key terms and concepts: ratio-cut algorithm • ratio • set cardinality • ratio cut



Networks, graphs, and partitioning.

(a) A network containing circuit logic cells and nets.

(b) The equivalent graph with vertexes and edges. For example: logic cell D maps to node D in the graph; net 1 maps to the edge (A, B) in the graph. Net 3 (with three connections) maps to three edges in the graph: (B, C), (B, F), and (C, F).

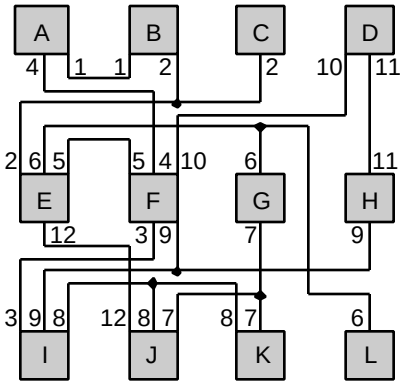
(c) Partitioning a network and its graph. A network with a net cut that cuts two nets.

(d) The network graph showing the corresponding edge cut. The net cutset in c contains two nets, but the corresponding edge cutset in d contains four edges. This means a graph is not an exact model of a network for partitioning purposes.

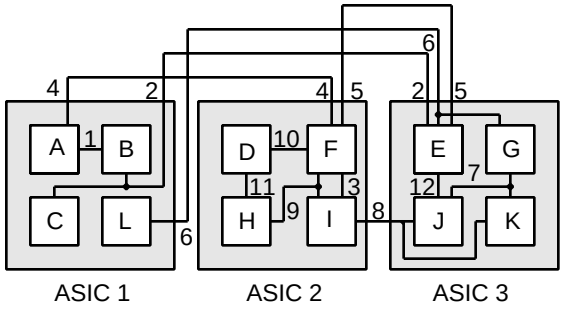
15.7.7 The Look-ahead Algorithm

Key terms and concepts: gain vector • look-ahead algorithm

(a)



(b)



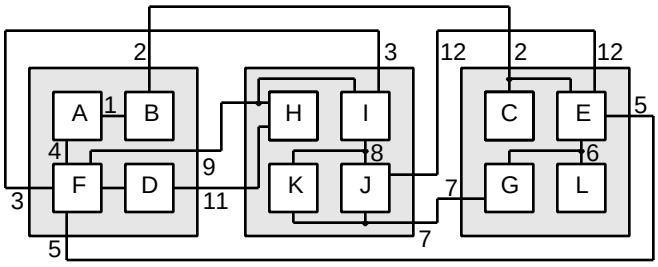
Partitioning example.

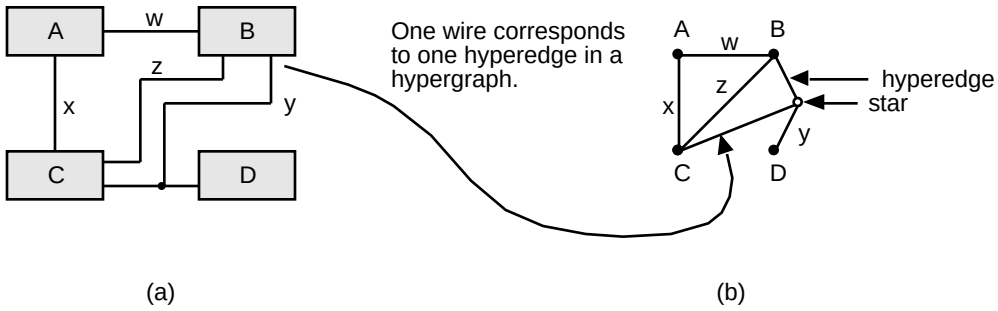
(a) We wish to partition this network into three ASICs with no more than four logic cells per ASIC.

(b) A partitioning with five external connections (nets 2, 4, 5, 6, and 8)—the minimum number.

(c) A constructed partition using logic cell C as a seed. It is difficult to get from this local minimum, with seven external connections (2, 3, 5, 7, 9,11,12), to the optimum solution of b.

(c)



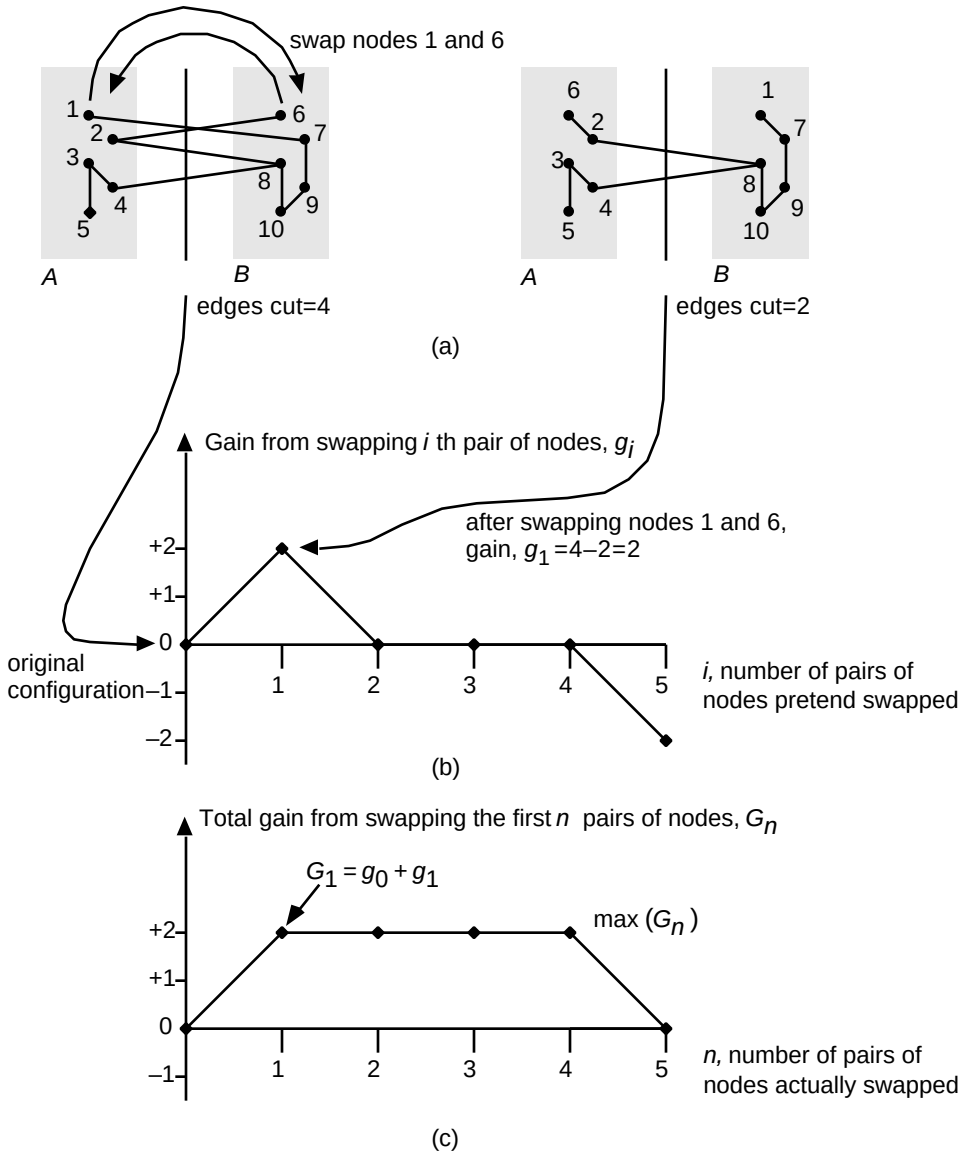


A hypergraph.

(a) The network contains a net y with three terminals.

(b) In the network hypergraph we can model net y by a single hyperedge (B, C, D) and a star node.

Now there is a direct correspondence between wires or nets in the network and hyperedges in the graph.

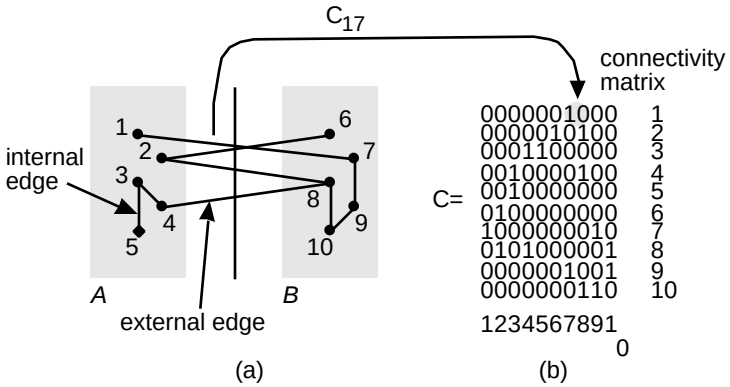


Partitioning a graph using the Kernighan–Lin algorithm.

(a) Shows how swapping node 1 of partition A with node 6 of partition B results in a gain of g=1.

(b) A graph of the gain resulting from swapping pairs of nodes.

(c) The total gain is equal to the sum of the gains obtained at each step.



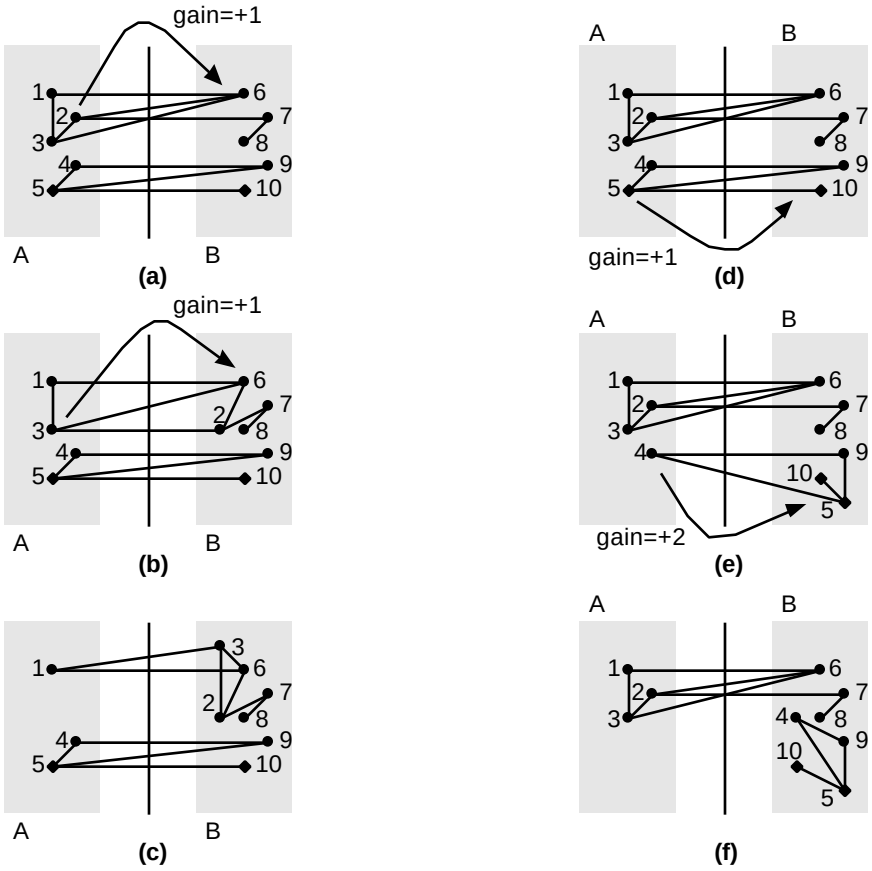
Terms used by the Kernighan–Lin partitioning algorithm.

(a) An example network graph.

(b) The connectivity matrix, C; the column and rows are labeled to help you see how the matrix entries correspond to the node numbers in the graph.

For example, C_{17} (column 1, row 7) equals 1 because nodes 1 and 7 are connected.

In this example all edges have an equal weight of 1, but in general the edges may have different weights.



An example of network partitioning that shows the need to look ahead when selecting logic cells to be moved between partitions.

Partitionings **(a)**, **(b)**, and **(c)** show one sequence of moves, partitionings **(d)**, **(e)**, and **(f)** show a second sequence.

The partitioning in **(a)** can be improved by moving node 2 from A to B with a gain of 1.

The result of this move is shown in **(b)**.

This partitioning can be improved by moving node 3 to B, again with a gain of 1.

The partitioning shown in **(d)** is the same as **(a)**.

We can move node 5 to B with a gain of 1 as shown in **(e)**, but now we can move node 4 to B with a gain of 2.

15.7.8 Simulated Annealing

Key terms and concepts: simulated-annealing algorithm uses an energy function as a measure • probability of accepting a move is $\exp(-\Delta E/T)$ • ΔE is an increase in energy function • T corresponds to temperature • we hill climb to get out of a local minimum • cooling schedule • $T_{i+1} = \alpha T_i$ • good results at the expense of long run times • Xilinx used simulated annealing in one verion of their tools

15.7.9 Other Partitioning Objectives

Key terms and concepts: timing, power, technology, cost and test constraints • many of these are hard to measure and not well handled by current tools

15.8 Summary

Key terms and concepts: The construction or physical design of a microelectronics system is a very large and complex problem. To solve the problem we divide it into several steps: **system partitioning**, **floorplanning**, **placement**, and **routing**. To solve each of these smaller problems we need **goals** and **objectives**, **measurement metrics**, as well as **algorithms** and **methods**

• The goals and objectives of partitioning
• Partitioning as an art not a science
• The simple nature of the algorithms necessary for VLSI-sized problems
• The random nature of the algorithms we use
• The controls for the algorithms used in ASIC design