Gaussian Interconnections for On-chip Networks

Ramon Beivide

Universidad de Cantabria
Avenida de los Castros s/n. 39005 Santander, Spain
e-mail: mon@atc.unican.es

1 Abstract

We consider in this research a subfamily of dense degree four Circulant graphs, containing the maximum number of nodes for a given diameter. We will show how the Gaussian integers, or the subset of the complex numbers with both real and imaginary integer parts, constitutes the appropriate mathematical model to deal with these graphs. For this reason, we denote them as Gaussian networks. Gaussian networks can be successfully applied to the design and modeling of on-chip parallel systems. For the same number of nodes and links, their richer connectivity and lower diameter make Gaussian networks topologically superior to Tori for both individual and collective communications.

In real life we try to travel via Euclidean paths in order to reduce the distance between two geographical sites. If the same were possible for messages traveling in a square Mesh, their longest path would correspond to its maximum Euclidean distance (the diagonal), that is $\sqrt{2}\sqrt{N}$. We will show that it is possible to find a mesh-like graph that, by using wrap-around links, halves this maximum distance. Such topology has been partially studied in the past as the dense member of the class of the degree four Circulant graphs. Its diameter is $\left\lfloor \frac{\sqrt{N}}{\sqrt{2}} \right\rfloor$. Hence, messages traveling between the farthest nodes will use paths whose distances are bounded by half of the maximum Euclidean distance in a square of size $\sqrt{N}\sqrt{N}$.

We define a *Circulant graph* with N vertices and jumps $\{j_1, j_2, \ldots, j_m\}$ as an undirected graph in which each vertex n, $0 \leq n \leq N-1$, is adjacent to all the vertices $n \pm j_i \ mod \ N$, with $1 \leq i \leq m$. We denote this graph as $C_N(j_1, j_2, \ldots, j_m)$. Rings and complete graphs of any size are particular cases of Circulants. The vertex-symmetry of Circulants allows their analysis starting from any vertex (node zero unless any other is stated), which simplifies their study. As the degree of a $C_N(j_1, j_2)$ graph is four, there can be, at most, $4d$ different nodes at distance d from node 0. Thus, for a given diameter k the maximum number of nodes of a $C_N(j_1, j_2)$ graph is:

$$N = 1 + 4\sum_{d=1}^{k} d = 1 + 4\left(\frac{k(k+1)}{2}\right) = 2k^2 + 2k + 1 = k^2 + (k+1)^2 \quad (1)$$

It is known that $C_N(k, k+1)$ graphs are dense Circulants of diameter k. Figure 1 shows one of these dense graphs for $k = 3$ and $N = 25$. We will see how dense Circulants can be viewed as bi-dimensional meshes with wrap-around links in which nodes are labeled by means of their integer coordinates in the plane. To this aim we employ a subset of the Gaussian integers for labeling the network nodes. The *Gaussian integers* $\mathbb{Z}[i]$ is the subset of the complex numbers with integer real and imaginary parts, that is, $\mathbb{Z}[i] := \{x + yi|\ x, y \in \mathbb{Z}\}$. Now, we define Gaussian networks.

Definition 1 *Given* $\pi = k + (k+1)i \in \mathbb{Z}[i]$ *we define the graph* $G_\pi = (V, E)$ *where:*

i) $V = \mathbb{Z}[i]_\pi$ *is the node set, and*

ii) $E = \{(\alpha, \beta) \in V \times V \mid (\beta - \alpha) \equiv \pm 1, \pm i \bmod \pi\}$ *is the edge set.*

We call G_π *the* dense Gaussian network *generated by* π*.* $C_N(k, k+1)$ *and* $G_{k+(k+1)i}$ *are isomorphic graphs.*

Figure 2 shows a G_{3+4i} which is isomorphic to $C_{25}(3, 4)$ shown in Figure 1. It can be seen that node zero (0,0) is located in the center of the network and the remaining nodes are labeled by Gaussian integers according to their integer coordinates in the complex plane. Hence, nodes in a dense Gaussian network will be labeled by all the possible (x, y) integer pairs such that $|x| + |y| \leq k$. Observe the presence of wrap-around links for maintaining the graph connectivity.

There are many applications over Gaussian networks that can benefit from a bi-dimensional labeling of the nodes. We will consider a number of them: unicast packet routing for an easy hardware implementation(Figures 3 and 4 and Algorithm 1 in the Appendix); perfect placement of resources (Figure 5); broadcast packet routing, also easily implementable in hardware (Figure 6 and Algorithm 2 in the Appendix); adaptive routing and deadlock-avoidance mechanisms; fault-tolerant routing; embedding commonly used communication topologies in parallel algorithms: Meshes, Rings of different sizes, Trees and Hamiltonian circuits, among others; practicable VLSI layouts, expansibility issues and hierarchic networks.

We will discuss with the audience the suitability of these networks for implementing massively on-chip parallel systems. In addition, we will present some ideas about new schemes for maintaining cache coherency over these networks.

2 Appendix

$\Delta x := x' - x$; $\Delta y := y' - y$;
DO IN PARALLEL:

$$\begin{cases} \Delta x_0 := \Delta x - 0; \\ \Delta y_0 := \Delta y - 0; \end{cases}$$

$$\begin{cases} \Delta x_1 := \Delta x - k; \\ \Delta y_1 := \Delta y - (k+1); \end{cases}$$

$$\begin{cases} \Delta x_2 := \Delta x + 1; \\ \Delta y_2 := \Delta y - (2k+1); \end{cases}$$

$$\begin{cases} \Delta x_3 := \Delta x + (k+1); \\ \Delta y_3 := \Delta y - k; \end{cases}$$

$$\begin{cases} \Delta x_4 := \Delta x + (2k+1); \\ \Delta y_4 := \Delta y + 1; \end{cases}$$

$$\begin{cases} \Delta x_5 := \Delta x + k; \\ \Delta y_5 := \Delta y + (k+1); \end{cases}$$

$$\begin{cases} \Delta x_6 := \Delta x - 1; \\ \Delta y_6 := \Delta y + (2k+1); \end{cases}$$

$$\begin{cases} \Delta x_7 := \Delta x - (k+1); \\ \Delta y_7 := \Delta y + k; \end{cases}$$

$$\begin{cases} \Delta x_8 := \Delta x - (2k+1); \\ \Delta y_8 := \Delta y - 1; \end{cases}$$

END DO IN PARALLEL
$(\Delta Re, \Delta Im) := (x_i, y_i)$ such that $|\Delta x_i| + |\Delta y_i| \leq k$;

Algorithm 1: Unicast Routing Algorithm in Terms of Sums and Comparisons.

if $distance = k$ **then**
    $distance := distance - 1$;
    Send packet to N with $NSEW = 1010$ (NE Triangle);
    Send packet to W with $NSEW = 1001$ (NW Triangle);
    Send packet to S with $NSEW = 0101$ (SW Triangle);
    Send packet to E with $NSEW = 0110$ (SE Triangle);
end
if $distance = 0$ **then**
    $CONSUME$ packet;
end
if $0 < distance < k$ **then**
    $CONSUME$ packet;
    $distance := distance - 1$;
    - Incoming packets with $NSEW = 1010$:
        Forward to N;
        Forward to E with $NSEW = 0010$;
    - Incoming packets with $NSEW = 1001$:
        Forward to W;
        Forward to N with $NSEW = 1000$;
    - Incoming packets with $NSEW = 0101$:
        Forward to S;
        Forward to W with $NSEW = 0001$;
    - Incoming packets with $NSEW = 0110$:
        Forward to E;
        Forward to S with $NSEW = 0100$;
    - Incoming packets with $NSEW = 1000$ $(0100, 0010, 0001)$:
        Forward to N (S,E,W);
end

Algorithm 2: One-to-All Routing Algorithm.

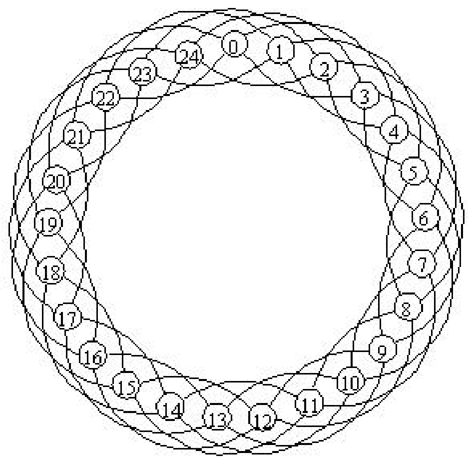

Figure 1: $C_{25}(3,4)$ Circulant Graph.

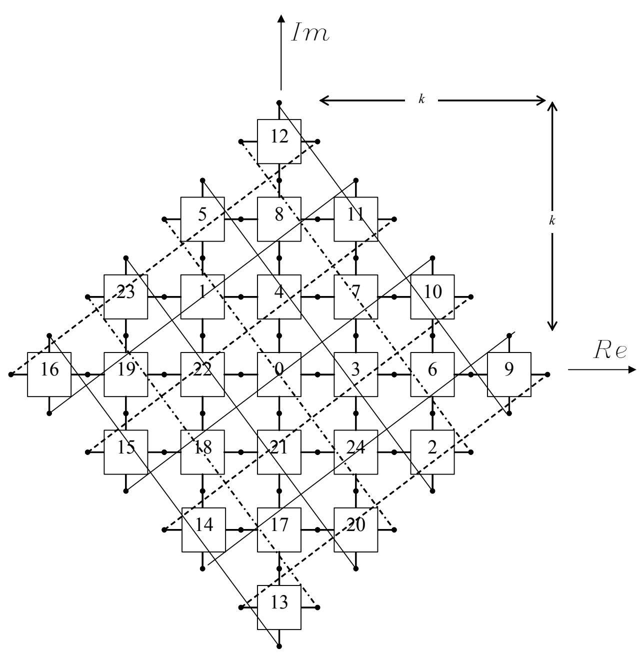

Figure 2: Gaussian network for $k = 3$.

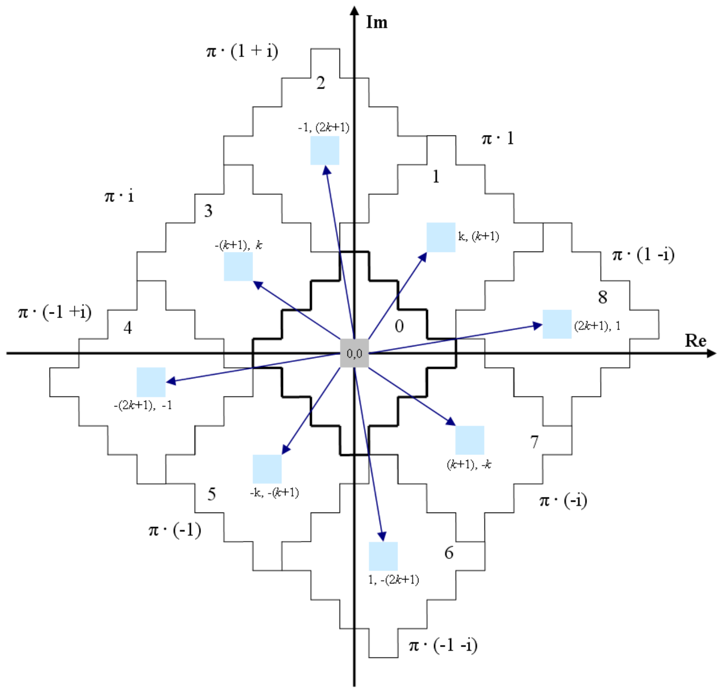

Figure 3: Unicast Routing Algorithm.

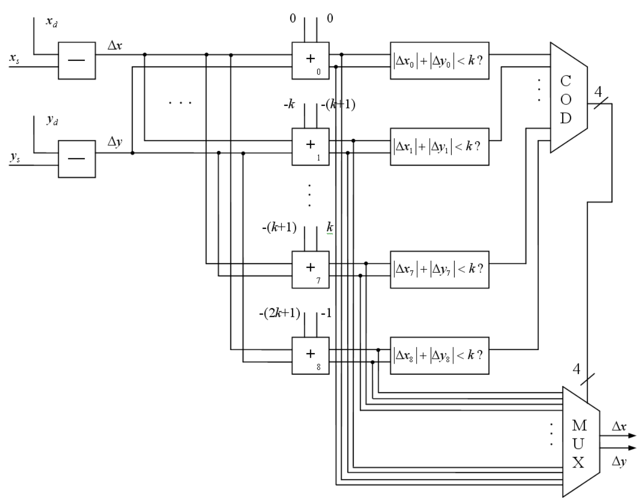

Figure 4: Routing Record Generator.

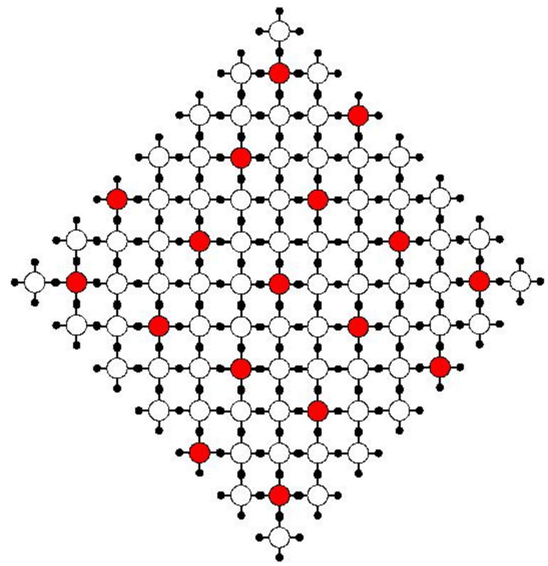

Figure 5: Perfect 1-Distance Placement in the Gaussian network of diameter 6.

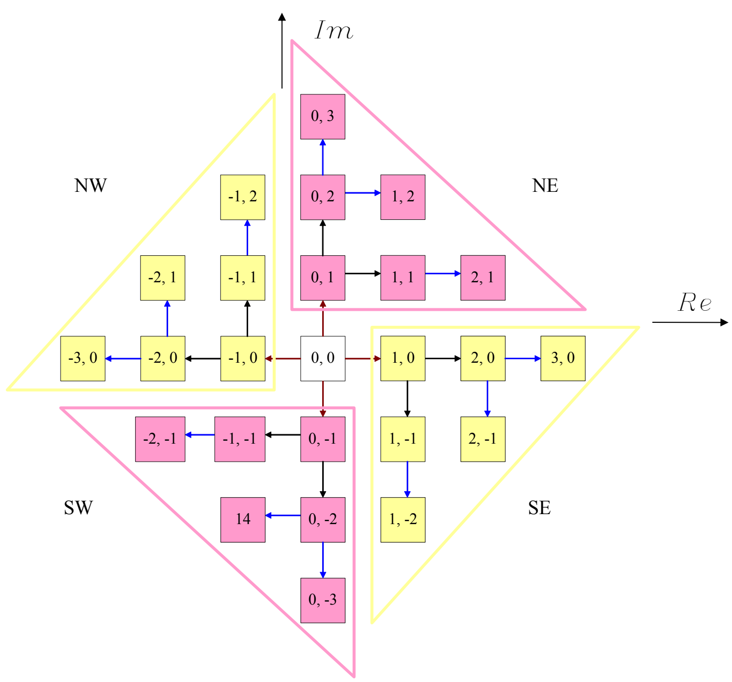

Figure 6: One-to-all Broadcast Algorithm.