A Global Routing Heuristic for FPGAs Based on Mean Field Annealing

İsmail Haritaoğlu and Cevdet Aykanat

Dept. of Computer. Eng & Information. Sci. Bilkent University 06533 Bilkent,
Ankara, TURKEY
hismail@bilkent.edu.tr

Abstract. In this paper, we propose an order-independent global routing algorithm for SRAM type FPGAs based on Mean Field Annealing. The performance of the proposed global routing algorithm is evaluated in comparison with LocusRoute global router on *ACM/SIGDA Design Automation* benchmarks. Experimental results indicate that the proposed MFA heuristic performs better than the LocusRoute in terms of the distribution of the channel densities.

1 Introduction

This paper investigates the routing problem in Static RAM (SRAM) based Field Programmable Gate Arrays (FPGAs) [7]. As the routing in FPGAs is a very complex combinatorial optimization problem, routing process can be carried out in two phases: *global routing* followed by *detailed routing* [5]. Global routing determines the course of wires through sequences of channel segments. Detailed routing determines the wire segment allocation for the channel segment routes found in the first phase which enables feasible switch box interconnection configurations [5, 9, 10].

Global routing in FPGA can be done by using global routing algorithms proposed for standard cells [5]. *LocusRoute* global router is one of this type of router used for global routing in FPGAs [4] which divides the multi-pin nets into two-pin nets and considers only two or less bend, minimum distance routes for these two-pin nets. The objective in LocusRoute is to distribute the connections among channels so that channel densities are balanced. In this work, we propose a new approach for the solution of global routing problem in FPGAs by using *Mean Field Annealing* (MFA) technique.

MFA merges collective computation and annealing properties of Hopfield neural networks [2] and simulated annealing [3], respectively, to obtain a general algorithm for solving combinatorial optimization problems [1]. MFA can be used for solving a combinatorial optimization problem by choosing a representation scheme in which the final states of the spins can be decoded as a solution to the target problem. Then, an energy function is constructed whose global minimum value corresponds to the *best solution* of the target problem. MFA is expected to compute the best solution to the target problem, starting from a randomly chosen initial state, by minimizing this energy function. Steps of applying MFA technique to a problem can be summarized as follows.

1) Choose a representation scheme which encodes the configuration space of the target optimization problem using spins. In order to get a good performance, number of possible configurations in the problem domain and the spin domain must be equal, i.e., there must be a one-to-one mapping between the configurations of spins and the problem.
2) Formulate the cost function of the problem in terms of spins, i.e., derive the energy function of the system. Global minimum of the energy function should correspond to the global minimum of the cost function.
3) Derive the mean field theory equations using this energy function, i.e., derive equations for updating averages (expected values) of spins.
4) Select the energy function and the cooling schedule parameters.

The FPGA model used in this paper are given in Section 2. The proposed formulation of the MFA algorithm for the global routing problem following these steps is presented in Section 3. The performance of the proposed MFA algorithm is evaluated in comparison with LocusRoute algorithm. Section 4 summarizes the implementation details of these two-algorithms. Finally, experimental results are presented in Section 5.

2 Global Routing Problem in FPGAs

The form of commercial FPGA consists of a two dimensional regular array of programmable logic blocks (LB's), a programmable routing network and switch boxes (SB's) [6, 13, 14]. Logic blocks are used to provide the functionality of a circuit. Routing network makes connections between LB's and input/output pads. Routing network of FPGA consists of wiring segments and connection blocks. Wiring segments have three type of routing resources in the commercial SRAM based FPGA [13]: channel segments, long lines and direct-interconnections. A horizontal (vertical) channel segment consists of a number of parallel wire segments connecting two successive SB's in a horizontal (vertical) channel. The SB's allow programmed interconnection between these channel segments. Direct-interconnection provides the connections between neighbor LB's. Long lines cross the routing area of FPGA vertically and horizontally. Connection blocks provide the connectivity from the input/output pins of LB's to the wiring segments of the respective channel segments. Each pin can be connected to a limited number of wiring segments in a channel and this is called as flexibility of connection block [7]. In this paper, it is assumed that each LB pin can be connected to all wiring segments in the respective channels. Therefore, we can omit the connection block in our FPGA model.

Since the *direct-interconnections* are used by neigbor LB's to provide minimum propagation delay and the *long lines* are used by signals which must travel long distances (i.e., global clock), these interconnection resources are not considered in the global routing. Hence, our FPGA model for global routing considers

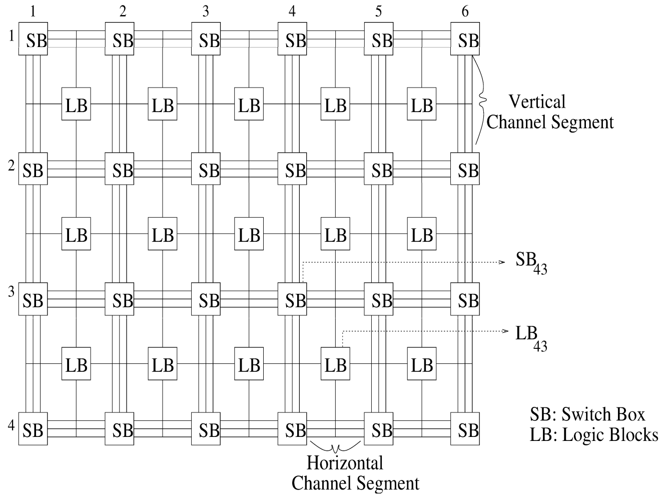


Fig. 1. The FPGA model used for Global Routing

only the LB's, SB's and channel segments. An FPGA can be modeled as a two dimensional array of LB's which are connected to the vertical and horizontal channel segments, and SB's which make connections between the horizontal and vertical channel segments (Fig. 1).

In this work, we divide all multi-pin nets into two-pin nets using minimum spanning tree algorithm [12] as in LocusRoute. Hence, a net refers to a two-pin net here, and hereafter. Consider the possible routings for a two-pin net with a Manhattan distance of $d_h + d_v$ where d_h and d_v denote the horizontal and vertical distances, respectively, between the two pins of the net on the LB grid. The routing area of this net is restricted to a $(d_h{+}1)\times(d_v{+}1)$ LB grid as shown in Fig. 2.a. Then, the shortest distance routing of this net can be decomposed into three *independent* routings as follows. Each pin of this net has only one neighbor SB in the optimal routing area. Hence, each pin can be connected to its unique neighbor SB either through a horizontal or a vertical channel segment (Fig. 2). Meanwhile, the optimal routing area for the connection of these two unique SB's is restricted to a $d_h \times d_v$ SB grid embedded in the LB grid (Fig. 2). Hence, by exploiting this fact, we further subdivide each net into three two-pin subnets referred here as LS, SS and SL subnets (Fig. 2.b). Here, LS and SL subnets represent the LB-to-SB and SB-to-LB connections, respectively, and SS subnets represent the SB-to-SB connection for a particular net. Therefore, we consider only two possible routings for both LS and SL subnets and $d_h{+}d_v{-}2$ possible one or two bend routings for SS subnets for routing the original net.

We define an FPGA graph $\mathbf{F}(L, S, C)$ for modeling the global routing problem in FPGAs. This graph is a $P \times Q$ two-dimensional mesh where L, S and C denote the set of LB's, SB's and channel segments, respectively. Here, P and Q

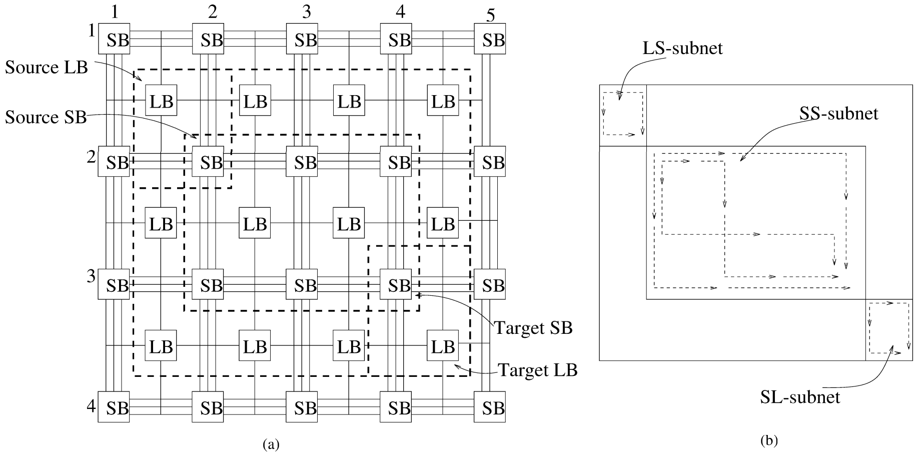


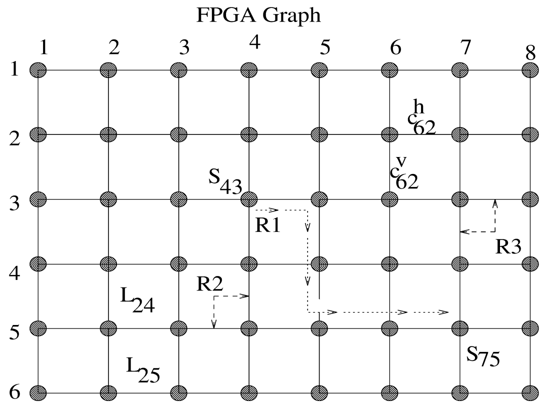
Fig. 2. (a) The routing area of the two-pin net and its subnets, (b) The possible routes for each subnets

is the number of horizontal and vertical channels in the FPGA. Each grid point (vertex) s_{pq} of the mesh represents the SB at horizontal channel p and vertical channel q. Each cell L_{pq} of the mesh represents the LB which is adjacent to four SB's s_{pq}, $s_{p,q+1}$, $s_{p+1,q+1}$ and $s_{p+1,q}$. Edges are labeled such that the horizontal (vertical) edge c^h_{pq} (c^v_{pq}) corresponds to the channel segment between the two consecutive SB's s_{pq} and $s_{p,q+1}$ ($s_{p+1,q}$) on the horizontal (vertical) channel p (q), respectively. Figure 3 displays a 8×6 sample FPGA graph. Then, the pins of the LS/SL and SS type subnets are assigned to the respective cell-vertex and vertex-vertex pairs of the graph as is in mentioned earlier.

The global routing problem reduces to searching for most uniform possible distribution of the routes for these subnets. The uniform distribution of the routes is expected to increase the likelihood of finding a feasible routing in the following detailed routing phase. Hence, we need to define an objective function which rewards *balanced* routings. We associate weights with the edges of FPGA graph in order to simplify the computation of the balance quality of a given routing. The weight w^h_{pq} (w^v_{pq}) of a horizontal (vertical) edge c^h_{pq} (c^v_{pq}) denotes the density of the respective channel segment. Here, the density of a channel segment denotes the total number of nets passing through that segment for a given routing. Using this model, we can express the balance quality B of a given routing $\mathbf{R}$ as

$$B(\mathbf{R}) = \sum_{p=1}^{P}\sum_{q=1}^{Q-1}(w^h_{pq}(\mathbf{R}))^2 + \sum_{q=1}^{Q}\sum_{p=1}^{P-1}(w^v_{pq}(\mathbf{R}))^2 \qquad (1)$$

As is seen in Eq. (1), each channel segment contributes the square of its density to the objective function thus penalizing imbalanced routing distributions. Hence,



R1: A possible route for SS-subnet (S_{43} , S_{75})

R2: Two possible routes for the LS-subnet (L_{34} ,S_{45})

R3: Two possible routes for the SL-subnet (S_{73} ,L_{73})

Fig. 3. The Cost Graph for FPGA model

the global routing problem reduces to the minimization of the objective function given in Eq. (1).

3 MFA Formulation

The MFA algorithm is derived by analogy to *Ising* and *Potts* models which are used to estimate the state of a system of particles, called spins, in thermal equilibrium. In Ising model, spins can be in one of the two states represented by 0 and 1, whereas in Potts model they can be in one of the K states. All LS/SL subnets are represented by Ising spins since they have only two possible routes. In Ising spin encoding of each LS/SL subnet m, $u_m = 1$ (0) denotes that the LB-to-SB or SB-to-LB routing is achieved through a single horizontal (vertical) channel segment. Each SS subnet n having $K_n \geq 2$ possible routes is represented by a K_n-state Potts spin. The states of a K_n-state Potts spin is represented using a K_n dimensional vector

$$\mathbf{v}_n = [v_{n1}, \ldots, v_{nr}, \ldots, v_{n,K_n}]^t \tag{2}$$

where "t" denotes the vector transpose operation. Each Potts spin $\mathbf{v}_n$ is allowed to be equal to one of the principal unit vectors $\mathbf{e}_1, \ldots, \mathbf{e}_r, \ldots, \mathbf{e}_{K_n}$, and can not take any other value. Principal unit vector $\mathbf{e}_r$ is defined to be a vector which has all its components equal to 0 except its r'th component which is equal to 1. Potts spin $\mathbf{v}_n$ is said to be in state r if $\mathbf{v}_n = \mathbf{e}_r$. Hence, a K_n-state Potts spin $\mathbf{v}_n$

is composed of K_n two state variables $v_{n1}, \ldots, v_{nr}, \ldots, v_{nK_n}$, where $v_{nr} \in \{0, 1\}$, with the following constraint

$$\sum_{r=1}^{K_n} v_{nr} = 1 \tag{3}$$

If Potts spin n is in state r (i.e., $v_{nr} = 1$ for $1 \leq r \leq K_n$) we say that the corresponding net n is routed by using the route r.

In the MFA algorithm, the aim is to find the spin values minimizing the energy function of the system. In order to achieve this goal, the average (expected) values $\langle u_m \rangle$ and $\langle \mathbf{v}_n \rangle = [\langle v_{n1} \rangle, \ldots, \langle v_{nr} \rangle, \ldots, \langle v_{nK_n} \rangle]^t$ of all Ising and Potts spins, respectively, are computed and iteratively updated until the system stabilizes at some fixed point. Note that for each Ising spin m, $u_m \in \{0, 1\}$, i.e., u_m can take only two values 0 and 1, whereas $\langle u_m \rangle \in [0, 1]$, i.e., $\langle u_m \rangle$ can take any real value between 0 and 1. Similarly, for each Potts spin n, $v_{nr} \in \{0, 1\}$ whereas $\langle v_{nr} \rangle \in [0, 1]$. When the system is stabilized, $\langle u_m \rangle$ and $\langle v_{nr} \rangle$ values are expected to converge to either 0 or 1 with the constraints $\sum_{r=1}^{K_n} \langle v_{nr} \rangle = 1$ for the Potts spins.

In order to construct an energy function it is helpful to associate the following meaning to the values $\langle u_m \rangle$ for LS/SL subnets.

$$\begin{aligned} \langle u_m \rangle &= \mathcal{P}(\text{subnet } m \text{ is routed by using the horizontal channel segment}) \\ 1 - \langle u_m \rangle &= \mathcal{P}(\text{subnet } m \text{ is routed by using the vertical channel segment}) \end{aligned}$$

That is, $\langle u_m \rangle$ and 1–$\langle u_m \rangle$ denote the probabilities of finding Ising spin m at states 1 and 0, respectively. In other words, $\langle u_m \rangle$ and $1 - \langle u_m \rangle$ denote the probabilities of routing subnet m through a single horizontal and vertical channel segment, respectively. Similarly, for SS subnets represented with Potts spins

$$\langle v_{nr} \rangle = \mathcal{P}(\text{subnet } n \text{ is routed through route r}) \qquad \text{for} \quad 1 \leq r \leq K_n \tag{4}$$

That is, $\langle v_{nr} \rangle$ denotes the probability of finding Potts spin at state r for $1 \leq r \leq K_n$. In other words, $\langle v_{nr} \rangle$ denotes the probability of routing net n through route r. Here and hereafter, u_m and v_{nr} will be used to denote the respective expected values ($\langle u_m \rangle$ and $\langle v_{nr} \rangle$,respectively) for the sake of simplicity. Now, we formulate the total density cost of global routing problem as an energy term

$$E_B(\mathbf{U}, \mathbf{V}) = \sum_{p=1}^{P} \sum_{q=1}^{Q-1} [w_{pq}^h(\mathbf{U}) + w_{pq}^h(\mathbf{V})]^2 + \sum_{q=1}^{Q} \sum_{p=1}^{P-1} [w_{pq}^v(\mathbf{U}) + w_{pq}^v(\mathbf{V})]^2 \tag{5}$$

$$\text{where} \quad w_{pq}^h(\mathbf{U}) = \sum_{m \ni c_{pq}^h} u_m \quad \text{and} \quad w_{pq}^h(\mathbf{V}) = \sum_{n \ni c_{pq}^h} \sum_{r \in R_n, r \ni c_{pq}^h} v_{nr}$$

$$w_{pq}^v(\mathbf{U}) = \sum_{m \ni c_{pq}^v} (1 - u_m) \quad \text{and} \quad w_{pq}^v(\mathbf{V}) = \sum_{n \ni c_{pq}^v} \sum_{r \in R_n, r \ni c_{pq}^v} v_{nr}$$

where $\mathbf{U} = \{u_1, u_2, \ldots\}$ and $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots\}$ represent the sets of Ising and Potts spins corresponding to the LS/SL and SS subnets, respectively. For

LS/SL subnets, "$m \ni c_{pq}$" denotes "for each LS/SL subnet m whose pair of pins share the horizontal or vertical channel segment c_{pq}". For SS subnets "$n \ni c_{pq}$" denotes "for each SS subnet n whose routing area contains the horizontal and vertical channel c_{pq}". Furthermore, "$r \in R_n, r \ni c_{pq}$" denotes "for each possible route r of SS subnet n which passes through the horizontal or vertical channel segment c_{pq}". Here, $w_{pq}(\mathbf{U})$ and $w_{pq}(\mathbf{V})$ represent the ***probabilistic densities*** of the horizontal or vertical channel segment c_{pq} for the current routing states of LS/SL and SS subnets, respectively. Hence, $w_{pq}(\mathbf{U},\mathbf{V}) = w_{pq}(\mathbf{U})+w_{pq}(\mathbf{V})$ represents the total probabilistic density of horizontal or vertical channel segment c_{pq} for the overall current routing state.

Mean field theory equations, needed to minimize the energy function E_B, can be derived as

$$\begin{aligned}\phi_m(\mathbf{U},\mathbf{V}) &= E_B(\mathbf{U},\mathbf{V})|_{u_m=0} - E_B(\mathbf{U},\mathbf{V})|_{u_m=1} \\ &= -2\big[w^h_{pq}(\mathbf{U},\mathbf{V}) - w^v_{pq}(\mathbf{U},\mathbf{V}) - 2(u_m - 0.5)\big] \qquad (6) \\ &\quad \text{where} \quad c^h_{pq}, c^v_{pq} \in m\end{aligned}$$

for an Ising spin m and

$$\begin{aligned}\phi_{nr}(\mathbf{U},\mathbf{V}) &= E_B(\mathbf{U},\mathbf{V})|_{\mathbf{V}_n=0} - E_B(\mathbf{U},\mathbf{V})|_{\mathbf{V}_n=e_r} \qquad (7) \\ &= -2\Big[\sum_{c^h_{pq}\in r}(w^h_{pq}(\mathbf{U},\mathbf{V}) - v_{nr}) + \sum_{c^v_{pq}\in r}(w^v_{pq}(\mathbf{U},\mathbf{V}) - v_{nr})\Big] \\ &\quad \text{for} \quad 1 \le r \le K_n\end{aligned}$$

for a Potts spin n, respectively. Mean field values ϕ_m and ϕ_{nr} can be interpreted as the increases in the energy function $E_B(\mathbf{U},\mathbf{V})$ when Ising and Potts spins m and n are assigned to states 1 and r, respectively. Hence, $-\phi_m$ and $-\phi_{nr}$ may be interpreted as the decreases in the overall solution qualities by routing LS/SL and SS subnets m and n through the horizontal channel and route r, respectively. Then, u_m and v_{nr} values are updated such that probabilities of routing subnets m and n through horizontal channel and route r increase with increasing mean field values ϕ_m and ϕ_{nr} as follows:

$$u_m = \frac{e^{\phi_m/T}}{1+e^{\phi_m/T}} \qquad (8)$$

$$v_{nr} = \frac{e^{\phi_{nr}/T}}{\sum_{k=1}^{K_n} e^{\phi_{nk}/T}} \quad \text{for} \quad r = 1, 2, \ldots, K_n \qquad (9)$$

respectively. After the mean field equations (Eqs. (6-7)) are derived, the MFA algorithm can be summarized as follows. First, an initial high temperature spin average is assigned to each spin, and an initial temperature T is chosen. Each u_m value is initialized to $0.5 \pm \delta_m$ and each v_{nr} value is assigned to $1/K_n \pm \delta_{nr}$ where δ_m and δ_{nr} denote randomly selected small disturbance values. Note that $\lim_{T\to\infty} u_m = 0.5$ and $\lim_{T\to\infty} v_{nr} = 1/K_n$. In each MFA iteration, the mean field effecting a randomly selected spin is computed using either Eq. (6) or Eq. (7). Then, the average of this spin is updated using either Eq. (8) or Eq. (9). This process is repeated for a random sequence of spins until the system is stabilized for the current temperature. The system is observed after each spin update in order to detect the convergence to an equilibrium state for a given temperature. If energy function E_B does not decrease in most of the successive spin updates, this means that the system is stabilized for that temperature. Then, T is decreased according to a cooling schedule, and iterative process is re-initialized. At the end of this cooling schedule, each Ising spin m is set to state 1 if $u_m \geq 0.5$ or to state 0, otherwise. Similarly, maximum element in each Potts spin vector is set to 1 and all other element are set to 0. Then, the resulting global routing is decoded as mentioned earlier.

4 Implementation

The performance of the proposed MFA algorithm for the global routing problem is evaluated in comparison with the well-known LocusRoute algorithm [4].

The MFA global router is implemented efficiently as described in Section 3. Average of each Ising spin m is initialized by randomly selecting u_m^{init} in the range $0.45 \leq u_m \leq 0.55$. Similarly, average of each Potts spin n is initialized by randomly selecting K_n v_{nr} values in the range $0.9/K_n \leq v_{nr} \leq 1.1/K_n$ and normalizing $v_{nr}^{init} = v_{nr}/\sum_{k=1}^{K_n} v_{nk}$ for $r = 1, 2, \ldots, K_n$. Note that random selections are achieved by using uniform distribution in the given ranges.

The initial temperature parameter used in mean field computation is estimated using the initial spin averages values. Selection of initial temperature parameters T_0 is crucial to obtain good routing. In previous applications of MFA, it is experimentally observed that spin averages tend to converge at a critical temperature. Although there are some methods proposed for the estimation of critical temperature, we prefer an experimental way for computing T_0 which is easy to implement and successful as the results of experiments indicate. We compute the initial average mean field as

$$\phi_{avg}^{init} = \Big(\sum_{m=1}^{N_m} \phi_m^{init} + \sum_{n=1}^{N_n}\sum_{k=1}^{K_n} \phi_{nr}^{init}\Big)/(N_m + \sum_{n=1}^{N_n} K_n)$$

Note that initial mean field values ϕ_m^{init} and ϕ_{nr}^{init} are computed according to Eqs. (6) and (7) using initial spin values u_m^{init} and v_{nr}^{init}. Here, N_m and N_n denote the total number of Ising and Potts spins, respectively, where $N = N_m + N_n$ denotes the total number of spins (subnets). Then, initial temperature is computed as $T_0 = C\phi_{avg}^{init}$ where constant C is chosen as 540 for all experiments.

The cooling schedule is an important factor in the performance of MFA global router. For a particular temperature, MFA proceeds for randomly selected unconverged net spin updates until $\Delta E < \epsilon$ for M consecutive iterations respectively where $M = N$ initially and $\epsilon = 0.05$. Average spin values are tested for convergence after each update. For an Ising spin m, if either $u_m \leq 0.05$ or $u_m \geq 0.95$ is detected, then spin m is assumed to converge to state 0 or state 1, respectively. For a Potts spin n, if $v_{nr} \geq 0.95$ is detected for a particular $r = 1, 2, \ldots, K_n$, then spin n is assumed to converge to state r. The cooling process is realized in two phases, slow cooling followed by fast cooling, similar to the cooling schedules used for Simulated annealing. In the slow cooling phase, temperature is decreased by $T = \alpha \times T$ where $\alpha = 0.9$ until $T < T_0/1.5$. Then, in the fast cooling phase, M is set to $M/2$, α is set to 0.8. Cooling schedule continues until 90% of the spins converge. At the end of this cooling process, each unconverged Ising spin m is assumed to converge to state 0 or state 1 if $u_m < 0.5$ or $u_m \geq 0.5$, respectively. Similarly, each unconverged Potts spin n is assumed to converge to state r where $v_{nr} = \max\{v_{nk} : k = 1, 2, \ldots, K_n\}$. Then, the result is decoded as described in Section 3, and the resulting global routing is found.

The LocusRoute algorithm is implemented as in [4]. As the LocusRoute depends on rip-up and reroute method, LocusRoute is allowed to reroute the circuits 5 times. No bend reduction has been done as in [6]. Both algorithms are implemented in the C programming language.

5 Experimental Results

This section presents experimental performance evaluation of the proposed MFA algorithm in comparison with LocusRoute algorithm. Both algorithms are tested for the global routing of thirteen *ACM SIGDA Design Automation* benchmarks (MCNC) on SUN SPARC 10 . The first 4 columns of Table 1 illustrate the properties of these benchmark circuits.

These two algorithms yield the same total wiring length for global routing since two or less bend routing scheme is adopted in both of them. Last six columns of Table 1 illustrate the performance results of these two algorithms for the benchmark circuits. The MFA algorithm is executed 10 times for each circuit starting from different, randomly chosen initial configurations. The results given for the MFA algorithm in Table 1 illustrate the average of these executions. Global routing cost values of the solutions found by both algorithms are computed using Eq. (1) and then normalized with respect to those of MFA. In Table 1, maximum channel density denotes the number of routes assigned to the maximally loaded channels. That is, it denotes the minimum number of tracks required in a channel for 100% routability.

As is seen in Table 1, global routing costs of the solutions found by MFA are 3.1%-10.5% better than those of LocusRoute. As is also seen in this table, maximum channel density requirements of the solutions found by MFA are less than those of LocusRoute in almost all circuits except *alu2* and *term1.* Both algorithms obtain the same maximum channel density for these two circuit.

Figures 4 and 5 contain visual illustrations as pictures (left) and histograms (right) for the channel density distributions of the solutions found by MFA and LocusRoute, respectively, for the circuit *C1355.* The pictures are painted such that the darkness of each channel increases with increasing channel density. Global routing solutions found by these two algorithms are tested by using SEGA [5] detailed router for FPGA. Figure 6 illustrates the results of the SEGA detailed router for the circuit *C1355*

Table 1. The performance results of the MFA and LocusRoute algorithms for the global routing of MCNC benchmark circuits

Benchmarks				Performance Results					
Circuits			FPGA size	MFA			LocusRoute		
name	number of nets	number of 2-pin nets		global routing cost	max. channel density	exec. time (sec)	global routing cost	max. channel density	exec. time (sec)
$9symml$	71	259	10x9	1.000	12.0	0.36	1.032	14	0.28
$too-large$	177	519	14x13	1.000	16.0	0.88	1.071	17	0.64
$apex7$	124	300	11x9	1.000	14.0	0.42	1.073	16	0.29
$example2$	197	444	13x11	1.000	15.0	0.64	1.097	16	0.72
vda	216	722	16x15	1.000	17.0	0.42	1.055	18	0.10
$alu2$	137	511	14x12	1.000	17.0	0.30	1.080	17	0.32
$alu4$	236	851	18x16	1.000	17.0	0.68	1.073	19	0.50
$term1$	87	202	9x8	1.000	14.0	0.34	1.093	14	0.27
$C1355$	142	360	12x11	1.000	13.0	0.56	1.119	15	0.43
$C499$	142	360	12x11	1.000	15.0	0.48	1.075	16	0.36
$C880$	173	427	13x11	1.000	15.4	0.68	1.065	17	0.38
$K2$	388	1256	21x19	1.000	20.2	0.94	1.038	22	0.60
$Z03D4$	575	2135	26x25	1.000	17.0	2.34	1.117	18	1.84

6 Conclusion

In this paper, we have proposed an order-independent global routing algorithm for FPGA, based on Mean Field Annealing. The performance of the proposed global routing algorithm is evaluated in comparison with the LocusRoute global router for 13 MCNC benchmark circuits. Experimental results indicate that the proposed MFA heuristic performs better than the LocusRoute.

7 Acknowledgments

The authors would like to thank Jonathan Rose for providing the benchmarks and necessary tools for FPGA.

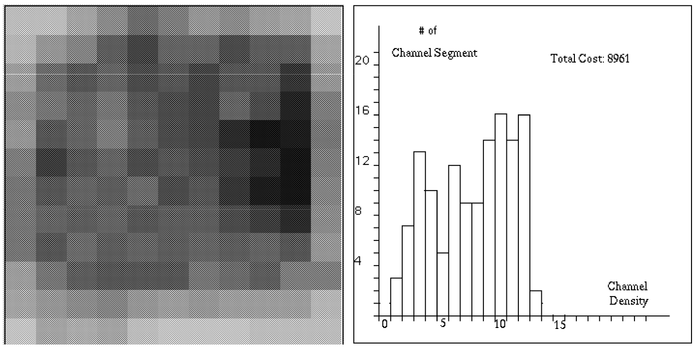


Fig. 4. Channel density distribution obtained by MFA for the circuit *C1355*

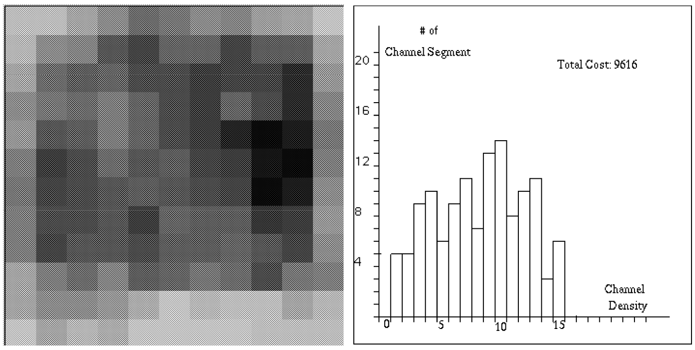
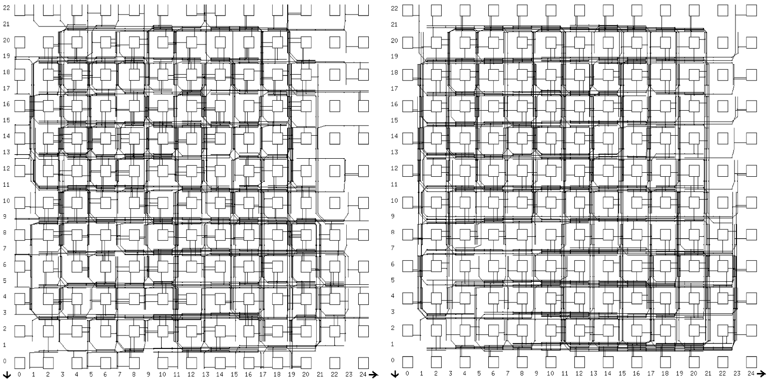


Fig. 5. Channel density distribution obtained by LocusRoute for the circuit *C1355*



(*a*) (*b*)

Fig. 6. SEGA detailed router results of the circuit *C1355* for the global routing solutions obtained by (a) MFA (b) LocusRoute

References

1. T. Bultan and C. Aykanat, " A new mapping heuristic based on mean field annealing, " *Journal of Parallel and Distributed Computing,* 16 (1992) 292-305.
2. J.J.Hopfield, , and D.W.Tank, "Neural Computation of Decisions in Optimization Problems", *Biolog. Cybern.,* vol. 52, pp. 141-152, 1985.
3. S.Kirkpatrick, C.D.Gelatt, and M.P.Vecchi. " Optimization by simulated annealing", *Science,* vol. 220, pp. 671-680, 1983.
4. J.Rose, "Parallel Global Routing for Standart Cells" *IEEE Transactions on Computer-Aided Design* Vol. 9 No. 10 pp:1085-1095 October 1990.
5. S.Brown, J.Rose, Z.Vranesic, "A Detailed Router for Field Programmable Gate Arrays" *Proc. International Conference on Computer Aided Desing 1990.*
6. J.Rose and B.Fallah, " Timing-Driven Routing Segment Assignment in FPGAs " *Proc. Canadian Conference on VLSI*
7. J.Rose, A. El Gamal, A. Sangiovanni-vincentalli, " Architecture of Filed-Programmable Gate Arrays " *Proceedings of the IEEE* pp:1013-1029 .Vol:81, No:7, July 1993.
8. C.Sechen, "VLSI Placement and Global Routing Using Simulated Annealing", *Kluwer Academic Publishers.* 1988
9. J.Greene, V.Roychowdhury, S.Kaptanoglu, A.E.Gamal, "Segmented Channel Routing", *27th ACM/IEEE Design Automation Conference* pp:567-572 1990.
10. S.Burman, C.Kamalanathan, N.Sherwani, "New Channel Segmentation Model and Routing for High Performance FPGAs", *Proc. International Conference on Computer Aided Desing, pp:22-25 1992*
11. N.Sherwani, "Algorithms for VLSI Physical Design Automation", *Kluwer Academic Publishers.* 1993
12. T.Lengauer, "Combinatorial Algorithms for Integrated Circuit Layout" 1990 *Wiley-Teubner Series.*
13. "Fundamentals of Placement and Routing", *Xilinx Co. 1990*
14. "The Programmable Gate Array Data Book", *Xilinx Co. 1992.*

This article was processed using the LaTeX macro package with LLNCS style