



Cellot Inc

Cellot

FPCL

Field Programmable Cell Logic

From Idea to Product & ASIC

FPCL Architectuer in Brief

CONFIDENTIAL

The information, data, drawings and the like contained in this document are proprietary to Cellot. The disclosure by Cellot of information contained herein does not constitute any license or authorization to use or disclose information, ideas or concepts presented. No part of this document may be disclosed or used for any purpose other than the review by the recipient. Information in this document is subject to change without notice and does not represent a commitment on the part of Cellot Inc.

FPCL Architecture

The FPCL is based on a collection of cells in a recursive, fractal like structure, which enables several different architectures. The first chosen architecture is a collection of registered memory cells and a matrix, which can route the output of each cell to the input of each cell and each I/O. Every cell is similar to any other cell. Any number of cells may be combined to create a larger similar cell. Each cell can be used for logic, implementing its memory as a look-up table, as storage or as both. The smallest (Atomic) Cell cannot be broken into smaller cells. Every Atomic Cell can be redundant to any other Atomic Cell whether it functions as a stand-alone application Cell or is part of a bigger cell.

Choosing different Atomic Cell sizes and their number on a chip allows the control of the chip's dimensions and other parameters. As an example: A device with 1M Bytes of memory (created by ~8000 1K varied data sized cells), 600 general-purpose I/O pins, 16 Global Clock inputs (8 of them having PLL for minimum skew) will provide a typical clock frequency of 250MHz (350 MHz max) and a die size of 1.7x1.7 Cm² (TSMC Library) in 0.13u (Standard Cells). Full custom design (in a slightly different architecture, which is presented below) will provide smaller size, and may double the frequency. The use of 1K memory Atomic Cells (varied in byte size to enable 1 bit data resolution) has been found as a good engineering trade-off. Small Atomic Cells increase the cross-connect wasting of "application area". The use of larger cells may become advantageous in future chip geometries and improved memories technology.

For full details, please refer to the patent publication:

European Patent No.: 1236222 (in US is still pending).

The following paragraph provides a brief explanation about Cellot patent.

The architecture concept

Different architectures may be built out according to the patent concept. This concept is about recursive architecture. Trying to combine few claims into one simplified (but not so accurate) concept:

"A universal hardware device comprising:

- a first plurality of cells for storing data; and

- a first programmable matrix connected to inputs and outputs of said at least one plurality of cells, whereby a plurality of hardware applications may be implemented by selectively storing data in said cells and selectively programming said matrix to connect at least one of said cells to at least one of said cells;

- wherein at least one of said cells comprises a second plurality of cells for storing data; and

- a second programmable matrix connected to inputs and outputs of said second plurality of cells, whereby each cell of said first plurality of cells and said second plurality of cells may have an architecture similar to the whole universal hardware device, and

- wherein the first programmable matrix and the second programmable matrix could be integral and.

- wherein at least some of the stored data is changeable dynamically by the hardware application during execution thereof".

A physical example for the upper demonstration is a cauliflower. Any flower out of the whole cauliflower may be considered as a whole cauliflower. This architecture may bring to mind the fractal structures ("in mathematics, any of a class of complex geometric shapes that commonly exhibit the property of self-similarity" [from the encyclopedia "*Britannica*"]).

An example to another cell architecture is the same cell without the register. Such a cell may work for applications that do not required feedback. Using the registers and the rest of the cells separately may give opportunity to combine synchronous and asynchronous implementations.

Full Custom Implementation

For full details, please refer to the patent publication.

The idea is to separate the chip into smaller devices. A 64K Bytes device is proposed here. As the amount of connections needed for connecting any point to any point in such a device is very small, a smaller delayed matrix will be used. Each device is implemented in a closed small area. Integrating the two parameters: smaller area and smaller delay in the cross connect, results in increased (about double) speed.

To connect the devices without degrading performance (compare to a non separated chip) another smaller delayed matrix is required. Therefore, the overall delay – compared to a non-separated chip – is increased (about 1:1.25), as the delay of two smaller delay matrixes is bigger than the delay of one original matrix.

Please note that only the original frequency and the doubled frequency are permitted, therefore none of the FPCL features is compromised in this implementation including the emulator, which retains the One-To-One feature. The Developer Environment is designed in such a way that its simulation speed decreases only to the parts for which the clock is doubled! As the work on the doubled speed device is not completed, it is safe to assume that the typical speed will be 2.5 nSec rather than the calculated speed (2.3 nSec).