Multi-purpose semi-static shift registers for digital programmable retinas

Thierry M. Bernard[*]

ENSTA/LEI , Paris, France
(Techniques Avancées / Laboratoire d'Electronique et d'Informatique)

Keywords: artificial retina, vision, array processor, SIMD architecture, VLSI, image processing.

ABSTRACT

A digital programmable retina is a functional extension of a CMOS imager, in which every pixel is fitted with a tiny digital programmable processor. We actually call it a PAR, standing for Programmable Artificial Retina. From an architectural viewpoint, a PAR is an SIMD array processor with local optical input. A PAR is aimed at processing images on-site (where they are sensed) until they can be output from the array under concentrated form. The overall goal is to get compact, fast and inexpensive vision systems, e.g. for robotics applications. PAR design is subject to harsh constraints resulting from small pixel area and sensing/processing cohabitation. Meeting these constraints leads to using peculiar architectural and circuit technique solutions. In the last three generations of PARs we have designed, semi-static shift registers have played a crucial role in the maximisation of computational power versus silicon area. In particular, the latter have been used to store, shift and — through some slight modifications — to perform local computations on images. Here, we show their abilities to support asynchronous propagation in order to implement "geodesic reconstruction", an extremely useful computational operator, in particular for image segmentation and then for object selection and manipulation purposes.

1. OUTLINE

Section 2 introduces programmable artificial retinas (PARs) in the context of CMOS imaging. Section 3 focuses on semi-static shift registers and particularly on their atomic component, the semi-static register, as most valuable building blocks in PAR design. Their progressive emergence over the last 20 years in several architectural aspects of PARs is shortly reviewed. Section 4 presents a novel semi-static-shift-register-like structure which features controllable propagation phenomena when appropriately operated. This property is to be exploited for the implementation of the so-called geodesic reconstruction operator and, more generally, to be able to address middle level vision with PARs

2. FROM CMOS IMAGERS TO PROGRAMMABLE ARTIFICIAL RETINAS

While pixel dimensions in focal plane arrays (FPAs) remain lowerbounded to a few microns because of hard optical limitations, CMOS transistor size keeps on decreasing in the so-called deep submicron range. Thus the continuing advances in VLSI technology will allow to lodge more and more transistors in the pixel — a few today, a few tens tomorrow — without significantly affecting sensing performances (array size, fill factor, noise...). Concurrent progress in thin film deposition, micro-optics and 3-D integration can only accelerate this evolution.

CMOS imagers are presently taking advantage of the new deal on the consumer market. A few transistors in each pixel are indeed enough to amplify the phototransduced signal in order to transfer it off the array through analog buses shared by rows of pixels. Image quality of CMOS imagers now matches that of low-end CCDs.

However, there is more to do with MOS transistors in the focal plane: they may be used to format or to process information. In fact, the use of CMOS technology allows to dramatically increase functional integration in the focal plane, with two major application-oriented motivations:

- easing the transmission of images; an exemplary case is the digital on-a-chip camera including A/D conversion, image formatting and compression (today's typical CMOS imagers only include ADC on-chip).

[*] tbernard@ensta.fr - www.etca.fr/~tbernard - Presented at Photonics West Conf. on Sensors, Cameras, and Systems for Scientific/Industrial Applications, SPIE Proc. 3965, San Jose, CA, January 24-25, 2000

- taking part in image analysis — within the frame of machine vision — in order to unburden/simplify the overall vision system and/or to react more quickly to external stimuli. Typical applications are found in robotics or man-machine interface.

Now, there are two ways to "smarten" the focal plane: either beside or inside the imaging array. One may speak of a *centralised* versus *distributed imager* approach.

The centralised solution amounts to gather an imager, an A/D converter and an image processing architecture on the same chip. There are already commercially successful products of this kind[1]. They actually come within a purely technological approach to vision system integration in the sense that the increased scale of integration is only exploited as a mean to move from PCB to chip level. The main motivations are to decrease the overall cost and to enlarge the communication bandwidths between the different parts. However, the information processing principles (representation, organisation, scheduling) remain similar. This approach is advantageous for product development because of its modularity and the ease of extrapolating future trends.

The distributed approach characterises imaging arrays in which there are transistors in each pixel that perform information processing. Because it implies a much deeper rapprochement between sensing and processing, technical and economical implications are much more difficult to appreciate. For the last ten years, we have explored this framework in which the imager actually becomes an *artificial retina*, borrowing their name to biological retinas as they intimately associate phototransducing devices with neural information processing structures. There are several fundamental motivations for the distributed approach:

- Working on the sensed data directly in the pixel allows the on-site concentration of information. For perception purposes, concentration may be all the more important that it is task-dependent. Then savings are expected not only in off-array and off-chip transmission but on all the rest of the vision system, with respect to complexity, energy, volume, weight, and then cost.
- Processing information where (and possibly as) it is transduced allows much more reactivity. It is not only useful for the observation of ultrafast phenomena. For example, it may allow to get rid of temporal aliasing artefacts that usually affect the time-sensitive computations required for motion analysis in natural sequences.
- Pixel intelligence is also the only solution for an individual control of each phototransducer. This can be used to finely adapt the sensitivity range in real-time. It could also allow to tune the spectral sensitivity or to accommodate (by local action on a micro-optical system), thus giving its full meaning to a concept of active sensing.

With about 10 transistors per pixel, it is already possible to perform some dedicated image preprocessing tasks, using analog circuit techniques[2]. However, we look for much more visual versatility. So we have been more attracted toward settling a programmable architecture in the focal plane[3]. We mean a 2-D SIMD array processor on a chip in which each processing element (PE) is closely connected to an elementary photosensitive device, thus corresponding to one pixel of the image. We call such a circuit a Programmable Artificial Retina (PAR). Ideally, a PAR should allow to perform all *retinotopic* operations needed in the visual process: those that transform images into other images. This concerns low and middle level vision, including early vision operators, image segmentation, pattern recognition... Because analog processing operators are rather specific — even within the powerful CNN paradigm[4] — and difficult to combine, we have naturally turned towards digital PARs to get the best versatility. This means that pixel-level ADC is needed prior to processing. Specific circuit techniques have been developed for this purpose[5,6].

3. SEMI-STATIC (SHIFT) REGISTERS FOR DIGITAL PARS

Considering the limited number of transistors affordable per pixel, the only meaning of "digital" here is that the data handled by the PE (processing element) are binary, not analog. There is not enough room for arithmetical operators. Only logical ones can be implemented. Furthermore, analog techniques are welcome if they provide compact solutions for handling the binary data.

There are five architectural issues in the design of a digital retina PE:

- data storage, in binary registers or local memory;
- data communication, within the PE (intra-pixel) or between neighbor PEs (inter-pixel);
- boolean and bit-serial computations to process binary and grey-level images;
- long distance communication;
- SIMD instruction decoding, if necessary.

It is instructive to review how the semi-static register shown on figure 1c has gradually emerged as a versatile building block bringing compact solutions to most of the above issues.

In the eighties, the first PARs were targeted at simple spatial binary image processing only. For them, binary image shifting was a crucial operator. Going back to the famous Mead & Conway's book[7], the dynamic shift register shown on figure 1a was well known as the simplest structure for enabling the unidirectional movement of a sequence of data. From this basic structure, more sophisticated ones could be obtained, such as the LIFO stack[7] shown on figure 1b, which allows bidirectional movement and storage of a list of data. Note that, in these structures, a single nMOS transistor is used as pass gate in spite of the signal degradation that occurs when transmitting a logical 1. While inconsequential in nMOS technology, this needs some care in CMOS technology. A simple solution[3] is to use a control voltage for the nMOS pass transistors slightly larger than the power supply voltage of inverters.

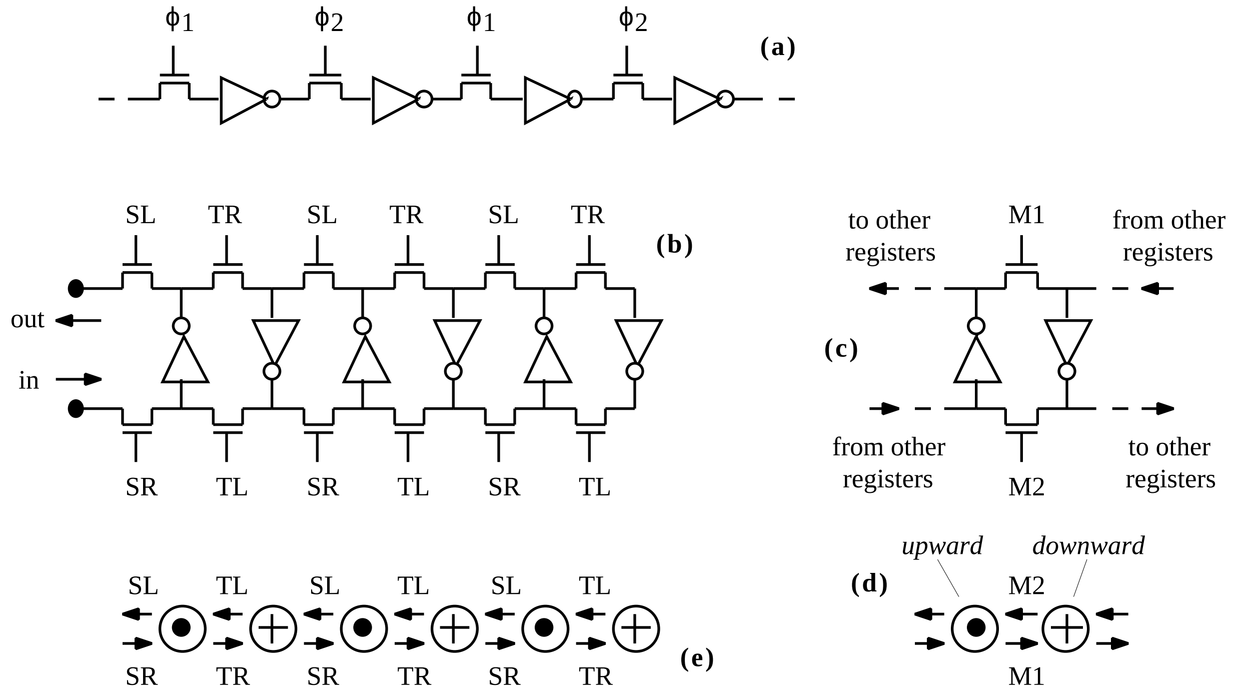

Figure 1: (a) Dynamic shift register (b) Binary LIFO stack of depth 3 obtained by interlacing two dynamic shift registers and sharing the inverters[7] (c) Semi-static register (d) Symbolic top view of the semi-static register (e) Symbolic top view of the stack.

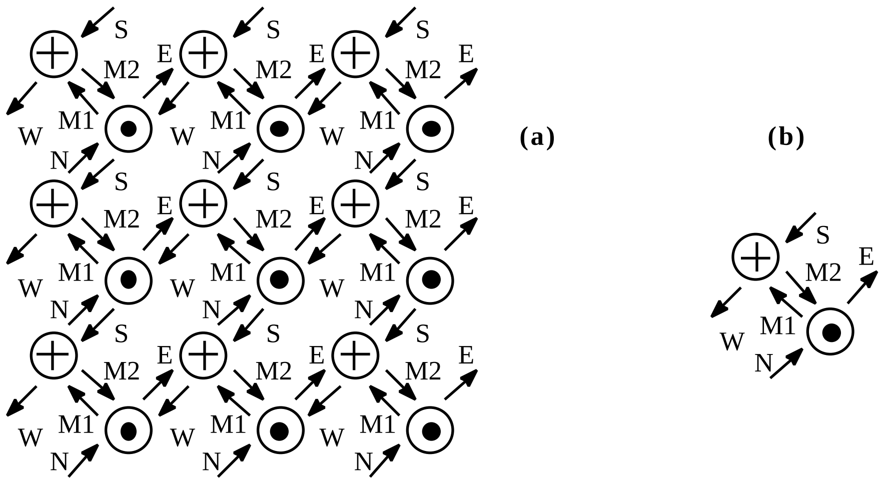

Figure 2: (a) Top view of a 2-D semi-static shifting structure represented using the conventions introduced on figure 1d&e. From a strict functional viewpoint, that kind of interconnection network is now often called a NEWS network[8]. In this spirit, the signals called N, E, W and S are respectively used to shift data to north, east, west and south, in combination with M1 and M2. (b) The elementary periodical cell of the NEWS network is the semi-static register introduced on figure 1c, plus 4 pass transistors used for connections with the 4 closest neighbors.

Extension of the stack principle to a two-dimensional shifting and storing structure is fairly straightforward. We show it on figure 2a using the notation introduced in figures 1d&e: each inverter is seen as a vertical arrow of which the top view shows either the point or the feathers, depending whether it is upward or downward. Pass transistors are represented using an arrow notation indicating the direction in which they transfer data. The structure shown on figure 2a was used to support binary image shifting in the first PARs[9]. Though it is not clear at first sight, the dimensional extension from 1-D to 2-D has an important conceptual consequence. In figure 1e, any inverter is a centre of symmetry of the structure. The inverter actually appears as the atomic element. This no longer holds in figure 2a. Symmetry only appears if one gathers inverters two by two, for example those connected by pass transistors controlled by M1 and M2, as suggested by figure 2b which represents the corresponding periodical cell. That cell is precisely a semi-static register (cf. figure 1c) bearing multiple connections to/from neighbor registers. Calling that register "semi-static" comes from the fact that it is equivalent to a static memory cell, except when M1 or M2 is reset. Given that M1 and M2 will be typically set and reset at frequencies in the MHz range, the level of safety is thus much closer to that of SRAM than that of DRAM. This is important in the context of imaging arrays which are possibly subject to parasitic photocurrents.

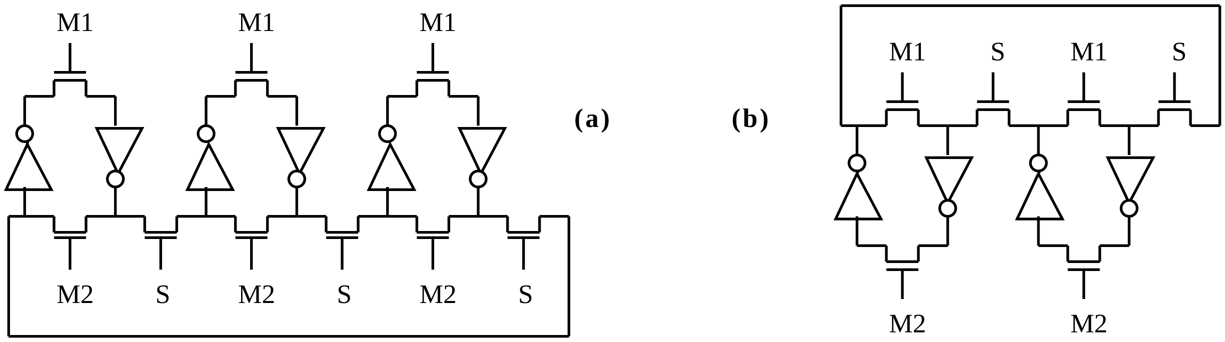

Figure 3: Small 1-D monodirectional ring semi-static shift registers. (b) As it links only two semi-static registers, this semi-static shift register is actually a swapper.

Using the semi-static register as an atomic element, there are various interesting "crystal" structures that can be devised[10]. Besides, small ring-connected semi-static shift registers, such as those shown on figure 3 prove to be useful at the pixel level: they provide a local memory structure with embedded read/write circuitry, with very few control signals and no need for a local address decoder (unlike RAM-based solutions). Note that control signals such as M1, M2, N, E, W, S in figure 2a form a multiphase clock system that generalises the two-phase system of the initial dynamic shift register of figure 1a. Then there is no instruction in the common sense to control these structures, only sequences of clock (phase) impulses. Using semi-static shift registers both for spatial image shifting and for local memory management allows the functional fusion[11] of inter- and intra-pixel communication facilities. In this context, a nice wire saving mechanism illustrated on figure 3 is that the same control signal S can be shared between two non intersecting shift registers, provided that S is validated by M1 and cancelled by M2 on the first one whereas it is validated by M2 and cancelled by M1 on the second one. We have extensively exploited these opportunities in various PAR designs[3,12,13]. They have proven very thrifty in terms of control hardware. With no instruction nor address decoder, and taking advantage of the various control signal sharing opportunities, a small overall number of global SIMD wires is sufficient for controlling each PE without wasting transistors in compensation[11].

Using semi-static registers instead of RAM cells for holding binary data in the pixel is also advantageous from a computational viewpoint. In particular, by resetting all control signals, all binary data and their complement come in charge form at the inverter inputs where they are stored. This makes it possible to exploit charge-domain analog computational operators. The most basic one is charge sharing as it just requires a single pass transistor connecting the inputs of two inverters. With matched capacitances, charge sharing computes the mean between the two initial voltages. Then using the inverters as asymmetric thresholders, the NAND or the NOR of the initial binary values is directly obtained[11]. So the incremental cost of such a NAND/NOR gate is just one transistor and one control signal.

As announced in the beginning of this section, semi-static registers have eventually proven to be valuable in nearly all architectural aspects of the PAR processing element. If one considers the ratio between memory capacity and technology-independent silicon area, designs based on semi-static registers[3,13] perform better[11] than designs based on a RAM organisation[5,14]. Note that, as only basic logical operators have been settled in PAR PEs so far, computational power is mainly dependent on PE memory capacity. However, there is one aspect that has remained unconcerned: that of long distance communication. Input/output to/from the outside of the array is generally better handled using column or row wires. But there are also long distance communication issues inside the array. The following section shows that the semi-static register approach also makes sense in this area.

4. GEODESIC RECONSTRUCTION FROM ASYNCHRONOUS PROPAGATION

4.1. Toward middle level vision

The ultimate goal of PARs is to carry out the whole retinotopic part of vision processes, in order to leave only a highly reduced number of concentrated and irregularly related pieces of information to an external computing resource such as a microcontroller. For most applications, reaching this goal means performing some middle level image processing on the PAR, most likely in close collaboration with the external microcontroller. At this level, computational objects are rather image regions than image pixels and a basic requirement is to be able to manipulate them easily within the PAR. A typical scenario would be that, given the address of a pixel (called a marker) by the microcontroller, the PAR would be able to quickly select all the other pixels that belong to the same region. This is illustrated on figure 4. Such an operation has been called a "geodesic reconstruction" and has proven a most fundamental algorithmic primitive[15].

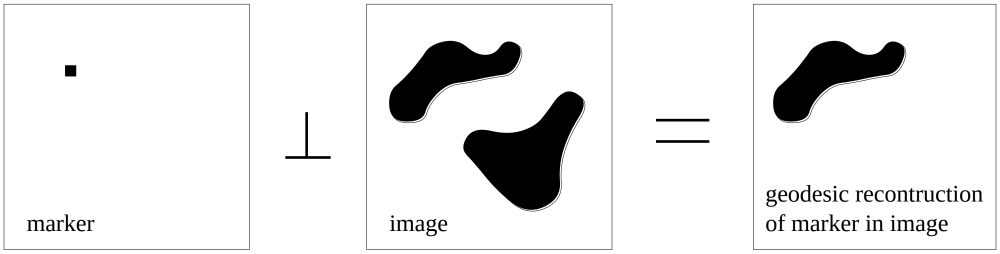

Figure 4: Illustrating geodesic reconstruction...

Geodesic reconstruction can be implemented though standard programming on PARs by iteratively performing pixel-to-pixel propagation. But the number of steps required is the so-called "geodesic diameter" of the reconstructed objects, which is always larger than the standard diameter. Furthermore, each step will take a few tens clock impulses. In fact, geodesic reconstruction has proven to be the most time consuming part in several applications. So it could be worth devoting some specific hardware to it if the extra cost is small. Another important motivation is power consumption reduction. A conservative digital solution has already been proposed[5] with a cost of about 20 transistors per pixel. We now present an analog solution that dramatically reduces that cost. The main originality lies in the autopropagation mechanism, to which section 4.2 and 4.3 are devoted.

4.2. Propagation principle

In order to both use minimal hardware and get maximal speed, we have looked for bidirectional propagation structures in which the propagating signal goes through the smallest number of gates per pixel. The structure we have come up with is shown on figure 5. Even if it resembles the stack structure shown on figure 1b, a major change is that here we will have one binary datum per inverter, an operating principle which is different from anything we have seen so far.

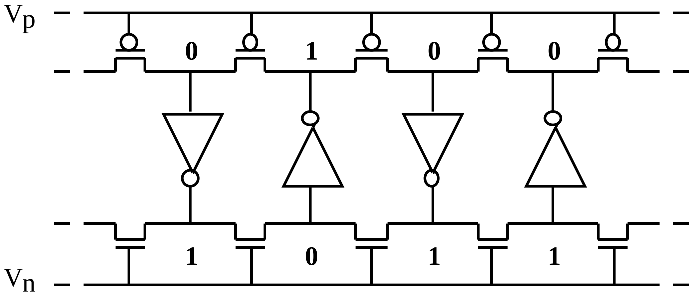

Figure 5: 1-D bidirectional propagation structure

This structure features one-dimensional (1-D) bidirectional propagation at small hardware cost. It actually corresponds to a row of pixels, with a single inverter per pixel. Each inverter may likely be part of a semi-static register in its pixel, but this is not represented on figure 5. The only important matter here is that, before triggering propagation, each inverter holds one binary datum from its pixel, which is dynamically stored on its input capacitance.

The structure also include horizontal transistors, that we call pass transistors though the nMOS transistors are meant to transmit only 0's and the pMOS transistors are meant to transmit only 1's. To prevent propagation, the Vp and Vn voltages are set such that these pass transistors are off.

For each inverter, let us consider its upper terminal, no matter whether it is the input or output of the inverter. When that upper terminal is set to 1 (respectively 0), the state of the inverter is denoted by ↑ (respectively ↓). These arrows somehow represent a vertical gradient. As an example, the structure of figure 5 is illustrated in state ··↓↑↓↓ ···. In the following, when turning on the pass transistors, the logical levels will become analog voltages. However, as is usual, we still call 0 (respectively 1) the input voltage of an inverter if it is significantly below (respectively above) the inverter threshold. Under such conditions, the ↑ and ↓ notations still make sense. Let Vss and Vdd be the ground and power supply voltages used by all inverters (transistor sources/wells/substrate).

Let us now describe the operating mode of the propagation structure. Initially, the control voltages of the pass transistors are set such that the pass transistors are off: Vn=Vss and Vp=Vdd. Let Vtn and Vtp be respectively the absolute values of the threshold voltages of nMOS and pMOS transistors. To trigger propagation, the pass transistors are turned slightly on thanks to the following changes:

Vn is increased from Vss to Vss+Vtn+Un

Vp is decreased from Vdd to Vdd-Vtp-Up

where Un and Up are small positive voltages tuned to put the pass transistors in strong enough inversion mode. As we will show, setting Un and Up has primarily to do with a trade-off between the robustness and speed of the propagation phenomenon. But let us first explain how propagation occurs. In the following explanations, we consider that subthreshold currents are negligible. This is not necessarily true but it allows to explain things more easily. We will also neglect the body effect for the same reasons.

With Vn = Vss+Vth+Un, the nMOS pass transistors are able to transmit voltages in the range [Vss , Vss+Un] but not above. Indeed, as soon as its source voltage is larger than Vss+Un, an nMOS pass transistor turns off. Let us now consider an upward inverter in state ↑ (such as the second inverter in figure 5, starting from the left). The inverter input will remain a 0, no matter which voltages are present on the other side of the connected nMOS pass transistors, if Vss+Un is smaller than the inverter threshold. Note that Un ≤ Vtn is a sufficient condition for inverters operated in strong inversion. Under such condition, ↑ is therefore a locally stable state for an upward inverter. By "locally stable", we mean that the inverter state is not affected by the spatio-temporal dynamics of the rest of the network (we are actually in the process of proving that this dynamics is of a propagation type).

Note that the present notion of stability has nothing to do with the fact that the inverter inputs are in dynamic state, except that the propagation phenomenon over the whole structure will have to be fast enough to be compatible with this dynamic storage. Anyway, we want it to be fast enough for computational purposes, which is an even stronger constraint.

Let us now consider a downward inverter in the same state ↑. Likewise, the pMOS pass transistors do not allow the input voltage of the inverter, initially set to 1, to decrease below Vdd-Up. So if Vdd-Up is higher than the inverter threshold, the inverter input will remain a 1. Under this condition, ↑ is also a locally stable state on a downward inverter. In summary, provided that Un and Up are small enough, state ↑ is always locally stable, whatever the direction of the inverter. Furthermore, choosing Un ≤ Vtn and Up ≤ Vtp satisfies this condition for inverters operated in strong inversion. It has the additional advantage that no significant static short-circuit current appears in the inverters, an important point for low power operation.

State ↑ is actually more than locally stable: it propagates. As we show now, any inverter in state ↓ will quickly turn over if one of its neighbor inverter is in state ↑. By "turning over", we mean it goes from state ↓ to state ↑. Let us consider the third inverter on figure 5 (starting from the left). As soon as Vp decreases to Vdd-Vth-Up, the pMOS pass transistors turn able to transfer logical level 1. As a consequence, the input voltage of the third inverter raises because it receives a 1 from the output of the second inverter. Meanwhile, the fourth inverter, which outputs a 0 because it is also in state ↓, cannot do anything to prevent that until the third inverter input voltage has reached Vdd-Up thus turning into a 1 which means that propagation has already occurred. As soon as the third inverter will have turned over, it will make the fourth inverter turn over too by propagating a 0 through the nMOS pass transistor. This dual procedure will repeat until all inverters are in state ↑. The latter situation corresponds to what we call "the global ↑ state" or "global state ↑" or, more shortly, "state —↑—", where sign "—" indicates repetition on both sides.

It might occur that no inverter is in state ↑. Then we say that the network is in "the global ↓ state" or, more shortly, "state —↓—". Note that —↓— is obviously stable because pass transistors are useless when they have the same voltage at their source and drain terminals. Actually, the latter are outside the passing range here. State —↓— is thus a dynamic (high impedance) stable state whereas state —↑— is a static one, driven by all inverters of the network.

In brief, once pass transistors slightly turned on, the spatio-temporal dynamics of the network leads to either of two possible stable states for the whole structure shown on figure 5: —↑— or —↓—. If one inverter is in state ↑ but not all are, that state propagates in both directions along the structure. The propagation speed from pixel to pixel must be roughly set by the time it takes for an inverter to (dis)charge the input of its neighbor through a pass transistor.

The 1-D propagation structure of figure 5 is easily extended to 2-D as illustrated by figure 6. The structure shown there corresponds to a 3x3 pixels subarray. Inverters are oriented following a checkerboard pattern such that the output of an arbitrary inverter is connected to the input of each of its closest neighbors through a single pass transistor, just as in the 1-D case. In 2-D, the hardware cost per pixel is 2 nMOS and 2 pMOS transistors, used as pass transistors. We do not count the inverter which is already part of the pixel processing element (PE). Note that some of the pass transistors might also be already used for other purposes in the PE, thus potentially reducing the incremental cost of settling propagation facilities in a PAR.

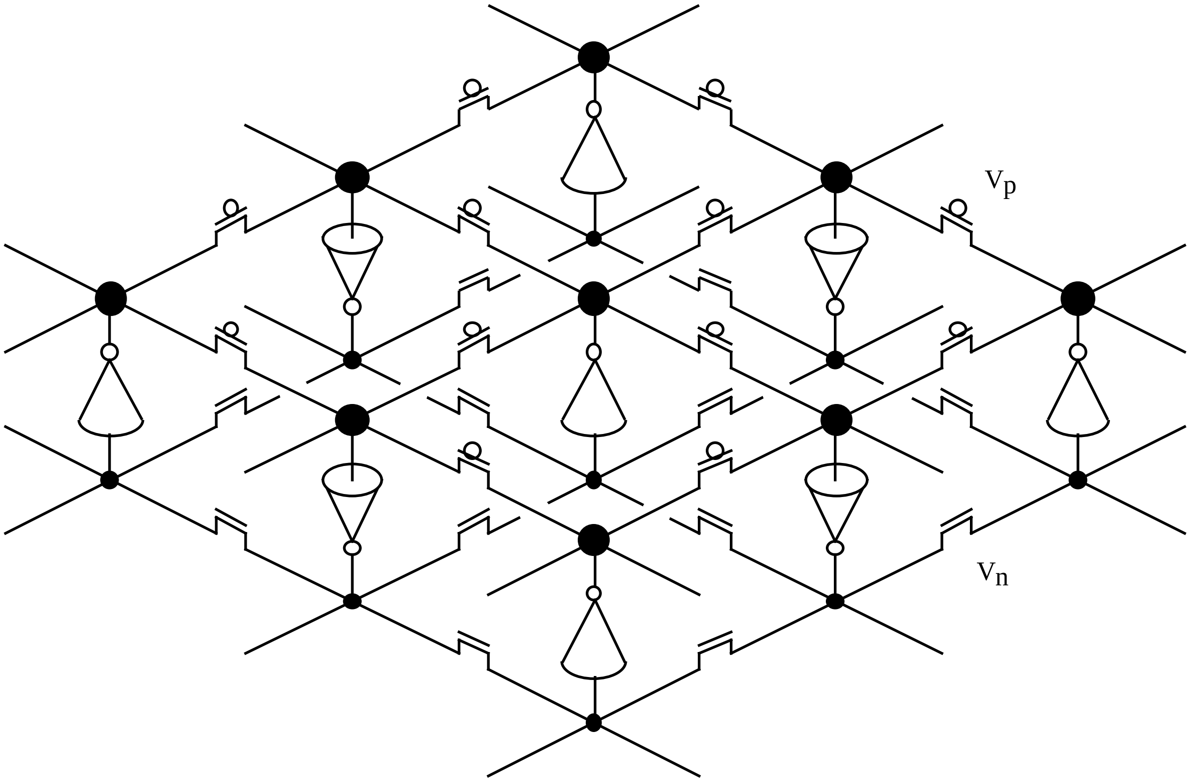

Figure 6: Perspective view of 2-D propagation structure. All nMOS and all pMOS pass transistors are respectively controlled by the Vn and Vp voltages.

4.3. Making the most of it

In the previous section, we have introduced a very compact structure which, under certain conditions, exhibits a propagation behavior. Now the point is to make the most of it. The present section provides some guidelines, with the support of simulation results. Note however that a precise analysis is beyond the scope of the paper.

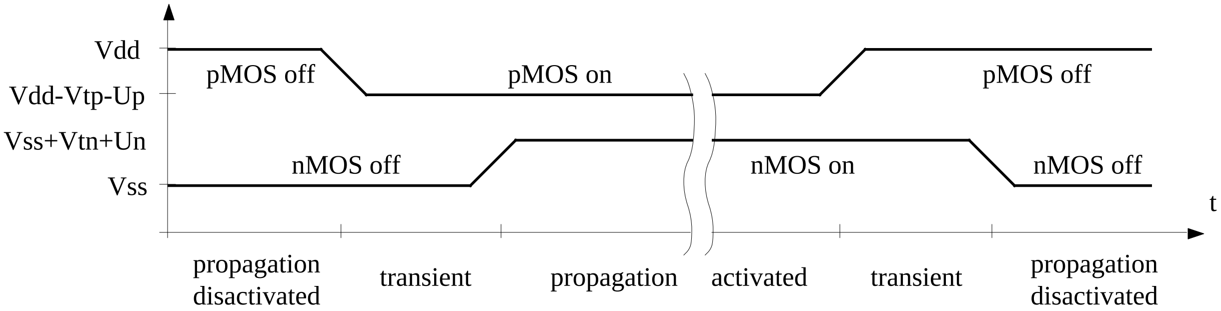

Figure 7: Turning propagation on and off.

To trigger propagation, Vn and Vp are both changed to turn all pass transistors on. However, it is difficult to assure that Vn and Vp will always change simultaneously in every pixel. The same problem exists when going back to the off state. Due to non simultaneity, the propagation network goes through transient control phases, as illustrated on figure 7, to which the propagation operation must remain insensitive. It actually does not matter whether propagation starts or not during these transient phases. The important point is that the local ↑ state (one inverter) and the global state —↓— (all inverters) both remain stable. The second condition is always met. As we show now, respecting the first condition imposes upperbounds on Un and Up.

Let us consider a single inverter in state ↑, surrounded by inverters in state ↓. If it is an upward inverter, the danger is that its input voltage, initially set to 0, increases beyond the inverter threshold voltage Vti due the 1's on the other side of the nMOS pass transistors. Neglecting body effect and subthreshold conduction (which are actually antagonistic), a necessary condition on Un to prevent the 0 from turning into a 1 is Vss+Un < Vti. Likewise, considering a lonely downward inverter in state ↑ leads to the following constraint: Vdd-Up > Vti. Expressing these constraints with respect to Vn and Vp yields:

$$Vn < Vti+Vtn \quad \text{and} \quad Vp > Vti-Vtp \qquad (1)$$

The larger Un and Up, the larger the currents through pass transistors and — most likely — the faster the propagation. It is thus tempting to set Vn=Vti+Vtn and Vp=Vti-Vtp, but it reduces the noise margin to zero which is dangerous. In practice, it is safer to set Vn=Vti+Vtn-ε and Vp=Vti-Vtp+ε with ε equal to a few tens of a volt, depending on the digital noise level in the circuit.

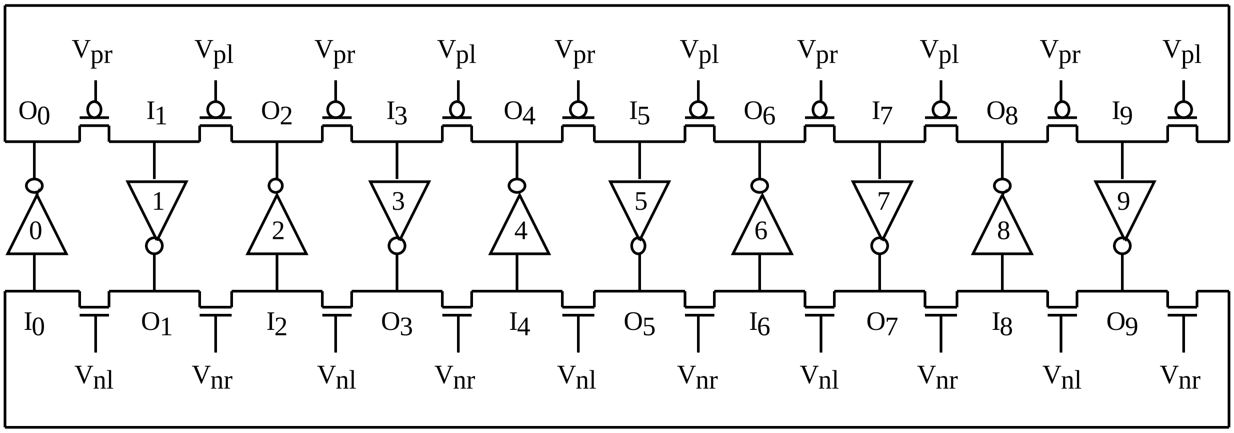

Figure 8: Simulated circuit

The circuit shown on figure 8 has been simulated to validate theoretical analysis. All transistors are chosen of minimal size, as they were in all structures we have mentioned in section 3. Doing so is a general way to reduce both area and power consumption. However it forces to design circuit techniques which are fairly independent of transistor strengths. But an important advantage is that such techniques scale well with technology shrinking.

Simulations were performed with BSIM3V3 transistor models corresponding to a 0.35µm standard CMOS technology. With BSIM3V3, transistors are continuously modelled from weak to strong inversion. By performing static DC analysis of the minimal size inverter operated at Vss=0V and Vdd=3.3V, it appears that Vtn=0.6V, Vtp=1.0V, and Vti=1.2V. Arbitrarily choosing ε=0.3V, equations (1) give Vn=1.5V and Vp=0.5V. On figure 8, Vn (respectively Vp) appears under the form of two different control voltages Vnl and Vnr (respectively Vpl and Vpr). This distinction will be useful later on. For the moment, Vn=Vnl=Vnr=1.5V and Vp=Vpl=Vpr=0.5V. On figure 8, each inverter is designated by its index k and its input and output are respectively called Ik and Ok. Each node is added a 10fF capacitance to realistically account for pixel-to-pixel wire interconnection.

Figure 9 shows the simulation results. The experiment carried out corresponds to the worst case of a single upward inverter in state ↑ (inverter 0) surrounded by 9 inverters in state ↓ (inverters 1 to 9). Note that the case of an upward inverter is worse than that of a downward inverter. Since inverters are built with minimal size inverters, it is indeed more difficult for them to impose a 1 than a 0 when fighting with their neighbors. In the present experiment, inverter 0 has to fight against inverters 2 and 8 to impose a 1. Actually, it may also feel the influence of inverters 4 and 6. In fact all inverters of odd index interact through the pMOS pass transistor chain as well as all inverters of even index interact through the nMOS pass transistor chain. To account for the certainly small but may be not negligible influence of inverters located more than two pixels away, it was more convincing to simulate a large enough network, hence the choice of a 10 inverter network. The ring structure also helps in this regard.

The evolution of the 20 nodes is shown on figure 9. To ease interpretation, they are separated into upper and lower nodes. If propagation is successful, all upper nodes will eventually be driven to Vdd, while all lower nodes will be driven to Vss. To improve readability, slightly different input voltage values have been used to initialise the transient analysis.

The results obtained are definitely successful. Propagation over the whole network takes a little more than 6ns. The slowest part corresponds to the beginning. Not only inverter 0 has to impose a 1 against inverters 2 and 8 on its output side, but also its input is attacked by inverters 1 and 9. This is why the I0 node voltage goes up to 0.7V. However, it remains significantly below the inverter threshold because of the safety margin resulting from ε. The fastest part of propagation corresponds to its end when the propagation wave reaches inverter 5 from both sides simultaneously (due to the ring structure). Probably the best place to measure propagation speed is between the steep falling edges of O1/O9 and O3/O7 observed on the lower chart. The time difference is about 3.0ns.

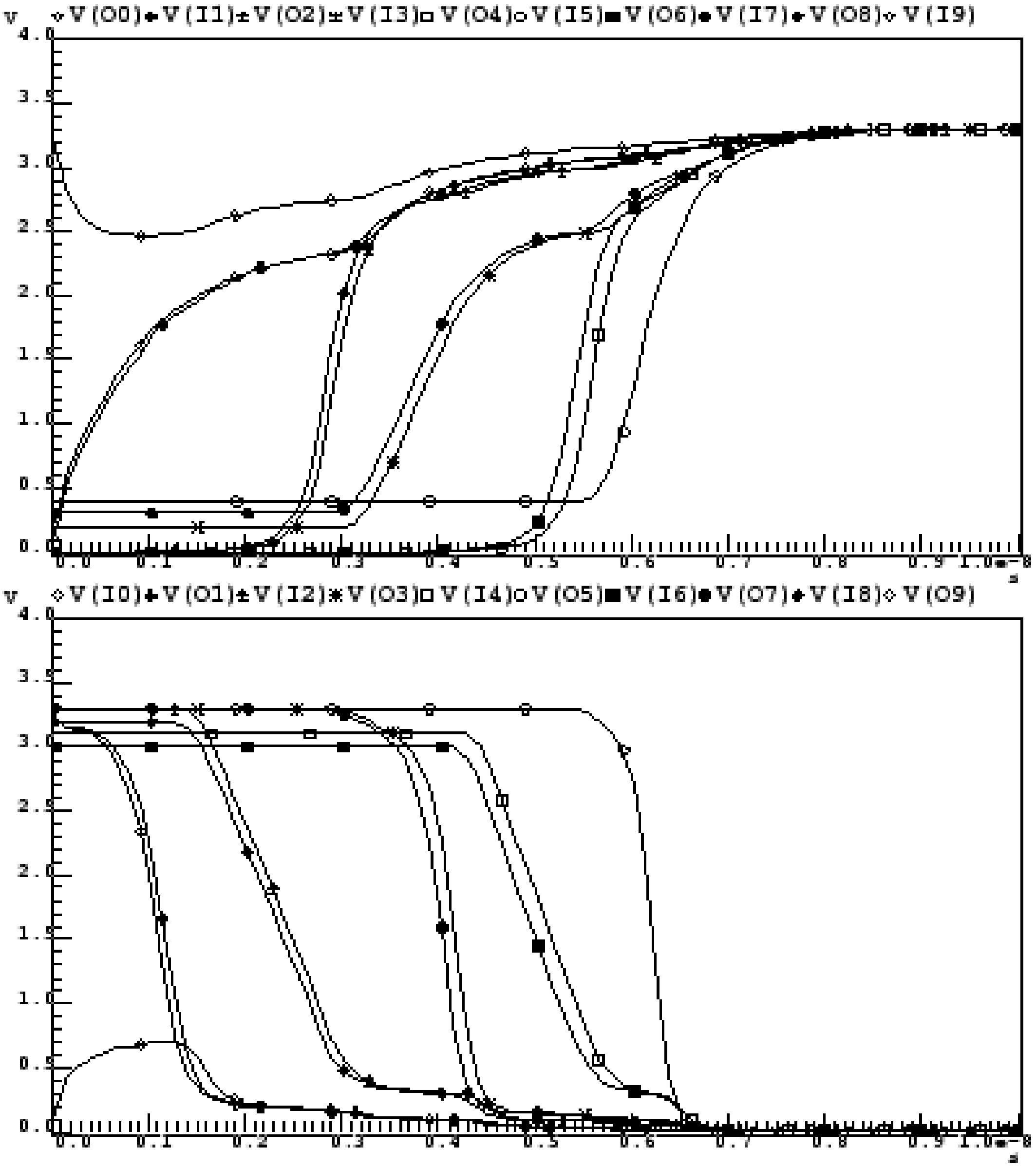

Figure 9: Simulation results for bidirectional propagation on the circuit of figure 8.

Finally, propagation time from pixel to pixel is about 1.5ns. This is about two orders of magnitude shorter than what could be done through synchronous programming on a PAR operated at 100MHz. The gain should become even larger for future large size PARs because technology shrinking makes the electromigration constraints more severe, with the eventual effect of limiting maximal clock rates in the PAR case. Just to take an example, on a PAR of size 240×320 pixels, somehow representing what could reasonably be manufactured in 0.35µm technology, propagation from one corner to the opposite corner of the array would take less than a 1µs. It is obviously compatible with the fact that data are only dynamically stored at the input of inverters.

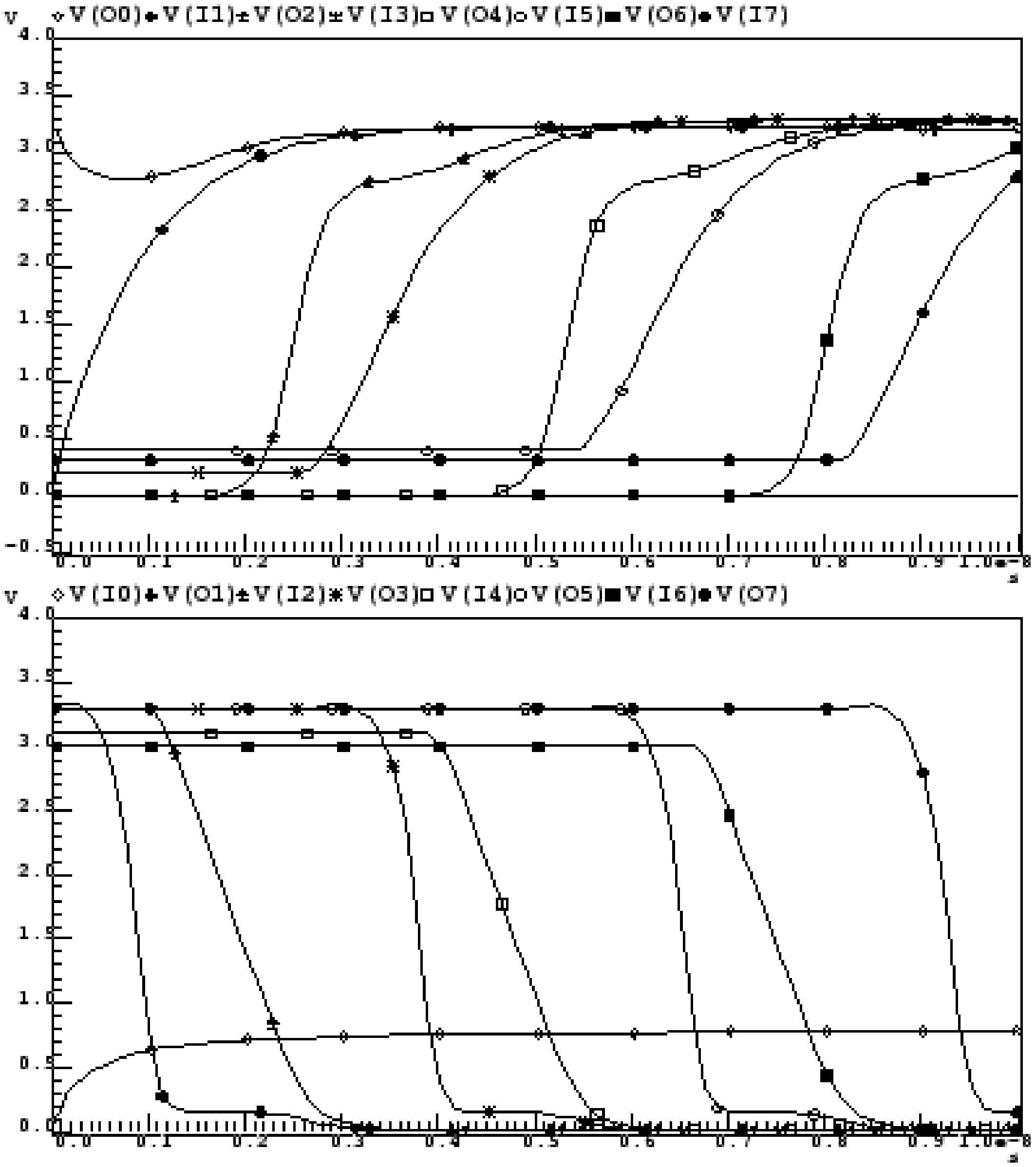

Figure 10: Simulation results for monodirectional propagation on the circuit of figure 8. Note that nodes I8, O8, I9 and O9 are not represented because their commutation occurs outside the 10ns window.

To better appreciate the propagation efficiency of the structure we have proposed, it is interesting to determine how fast monodirectional propagation would have occurred under the same operating conditions but without any fight between inverters. Obtaining such a behavior is simple with our structure. By turning off every other pass transistor, our propagation structure can be transformed into a simple inverter chain. On figure 8, the distinction between Vpr and Vpl, as well as between Vnr and Vnl, was made for this purpose. Turning on only the pass transistors controlled by Vnr and Vpr (respectively Vnl and Vpl) turns the inverter chain towards the right-hand side (respectively the left-hand side). For the simulation shown on figure 10, Vnr=1.5V and Vpr=0.5V while Vnl=0.0V and Vpl=3.3V. The obtained rightward inverter chain looks like the dynamic shift register shown on figure 1a, except that every other pass transistor is a pMOS one. But the main difference lies in the operating mode. Remember that there is one binary data per inverter here. And instead of alternatively activating ϕ1 and ϕ2 (cf. figure 1a), Vnr and Vpr are steady voltages during propagation, that partially turn on the pass transistors they control.

Simulation results presented on figure 10 show that monodirectional propagation speed is very close to bidirectional one. The time difference observed between the falling edges of O1 and O3 is about 3.0ns too. So the difference with the bidirectional case is insignificant. Such a result is actually no surprise, but it is "reassuring" to see theory confirmed by simulation. Respecting equations (1) with some additional ε margin essentially prevents inverters from fighting against each other: there is no time-wasting race between them. A race is a short-circuit between ground and power supply going through the nMOS transistor of an inverter and the pMOS transistor of another one. The absence of races is a boon. Not only it directly speeds up the propagation phenomenon, but it shows that energy consumption is limited to two items:

- charge and discharge of capacitances; the energy spent depends on Vdd-Vss (quadratically) and on the actual data, but not on the operating mode;
- dynamic short-circuit currents in inverters during voltage transitions; the energy spent grows linearly with the transition time.

Then, the faster the propagation speed, the less energy it takes. The propagation speed obviously depends on Vdd-Vss and on the ε margin. So, it can be concluded that, given Vdd-Vss, making the most of our propagation network is a trade-off between robustness, on the one hand, and either speed or energy, on the other hand. A precise analysis of this trade-off will be the subject of further investigation.

Another important remark concerns the functional interest of the monodirectional exploitation of our propagation structure. Because the circuit that we have simulated has a ring structure, its stability properties remain unchanged whether it is used for bidirectional or monodirectional propagation. In any case, there are only two global stable states: —↑— or —↓—. However, opening the ring, as will be done on a PAR, modifies the stability properties. If rightward propagation is triggered, there are new stable states representable by —↓↑—, we mean all sequences made of the concatenation of one subsequence of ↓ on the left hand side and one subsequence of ↑ on the right hand side. What happens at the transition ↓↑ is exactly what happens to inverter 0 on figure 10, before inverter 9 is reached by the propagation wave (this event is actually outside the time window presented). The input of inverter 0 is pulled up by inverter 9 until the pass transistor between them turns off because it source voltage is too high. It nearly reaches 0.8V after 10ns on figure 10, that is 0.4V below the inverter threshold, thus following the margin imposed by ε and by second order effects such as subthreshold conduction and body effect. Equations (1) obviously apply to that situation. Oriented propagation has some applications in image processing[16], so it is nice to get it as a byproduct of our propagation structure. Extrapolating to 2-D, note that an eastward propagation would leave one inverter per line — the first one in state ↑ starting from the left — in significant short-circuit. However, much more energy saving is expected from the asynchronous character of the propagation.

4.4. Geodesic reconstruction

Omnidirectional propagation has little interest in itself but to compute a global OR on a binary image. It is certainly most important to quickly get that type of information for a large class of image processing algorithms, but simple solutions already exist[17]. Propagation is really useful only if it can be conditioned by local binary data. To perform the geodesic reconstruction illustrated on figure 4, it is necessary to prevent propagation everywhere the "image" is white. To do this, every pass transistor connected to a pixel in the white part of the image should remain off. This can be easily done if there is a local control of each pixel on the distribution of voltages Vn and Vp to the pass transistors. Then geodesic reconstruction is a 3 step procedure:

a) initialise all inverters in state ↓, except those that correspond to the black pixels on the marker image, to be put in state ↑;

b) turn on all pass transistors contained within the black regions of the image, by following the methodology developed in the two previous sections for omnidirectional propagation (monodirectional or other intermediate forms also applies) ;

c) wait for propagation to terminate.

The result will be the right image shown on figure 4.

5. CONCLUSION

Beside all the advantages that semi-static (shift) registers have shown, drawbacks do exist. One of them is that the complex shifting structures evoked in section 3 make the control problem difficult. In particular, there is a severe lack of orthogonality (from the computer architecture viewpoint) between the different elementary actions which allow the SIMD control of processing elements. Also, shift-registers used as local memory structure tend to move (much) more data than what is actually needed by the ongoing computations. This leads to inefficiencies from both the computational power and energy consumption viewpoints. With technology scaling down, the number of pixels and the local memory size in the pixel will both grow, worsening the drawbacks we have just mentioned. Then one may wonder whether the advantages of semi-static shift registers, that have so much helped us pioneering PARs, will remain valuable in the future. The geodesic reconstruction operator presented in section 4, which is nothing else than a semi-static shift register operated in some peculiar fashion, suggests they might still have a long way to go.

6. REFERENCES

[1] M. Gökstorp and R. Forchheimer. Smart Vision Sensors. *Proc. of IEEE Int. Conf. on Image Processing*, pp. 479-482, 1998.

[2] C. Koch and H. Li, ed. *VISION CHIPS - Implementing Vision Algorithms with Analog VLSI Circuits*. IEEE Computer Society Press, 1995.

[3] T.M. Bernard, B.Y. Zavidovique, and F.J. Devos. A programmable artificial retina. *IEEE Journal of Solid-State Circuits*, 28(7):789-798, July 1993.

[4] L.O. Chua and L. Yang. Cellular neural networks: Theory - Applications. *IEEE Trans. on Circuits and Systems*, 35(10):1257-1290, October 1988.

[5] J.-E. Eklund, C. Svensson, and A. Astrom. VLSI implementation of a focal plane image processor - a realization of the near-sensor image processing concept. *IEEE Transactions on VLSI Systems*, 4(3):322-335, September 1996.

[6] D.X.D. Yang, B. Fowler and A. El Gamal. A Nyquist-Rate Pixel-Level ADC for CMOS Image Sensors. *IEEE Journal of Solid-State Circuits*, 34(3):348-356, March 1999.

[7] C. Mead and L. Conway. *Introduction to VLSI Systems*. Addison-Wesley, Reading, MA, 1980.

[8] D. Hillis. *The Connection Machine*. MIT Press, Cambridge, MA, 1986.

[9] P. Garda, B. Zavidovique and F. Devos. Integrated cellular array performing neighborhood combinatorial logic on binary pictures. *Proc. of IEEE ESSCIRC*, 1985.

[10] T. Bernard. *Des Rétines Artificielles Intelligentes*. PhD Thesis, Paris XI University, 1992.

[11] F. Paillet, D. Mercier, and T.M. Bernard. Making the most of 15k lambda2 silicon area for a digital retina PE. *Proc. SPIE, Vol. 3410, Advanced Focal Plane Arrays and Electronic Cameras*, 1998.

[12] R. Nguyen, D. Mercier, A. Jullian, and T.M. Bernard. Hardware-software aspects of shift register based NEWS networks for the focal plane. *Proc. Workshop on Computer Architecture for Machine Perception*, pp. 84-93, 1997.

[13] F. Paillet, D. Mercier and T.M. Bernard. Second Generation Programmable Artificial Retina. *Proc. IEEE ASIC Conference*, pp. 304-309, 1999.

[14] M. Ishikawa, K. Ogawa, T. Komuro and I. Ishii. A CMOS Vision Chip with SIMD Processing Element Array for 1ms Image Processing. *Proc. Int. Solid-State Circuit Conf.*, pp. 206-207, San Francisco, CA, 1999.

[15] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, UK, 1982.

[16] A. Astrom, R. Forchheimer, and J.E. Eklund. Global feature extraction operations for near-sensor image processing. *IEEE Transactions on Image Processing*, 5(1):102-110, January 1996.

[17] T.M. Bernard and P.E. Nguyen. Vision through the power supply of the NCP retina. *Proc. SPIE, Vol. 2415, Charge Coupled Devices and Solid State Sensors V*, pp. 159-163, 1995.