

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
8 February 2001 (08.02.2001)

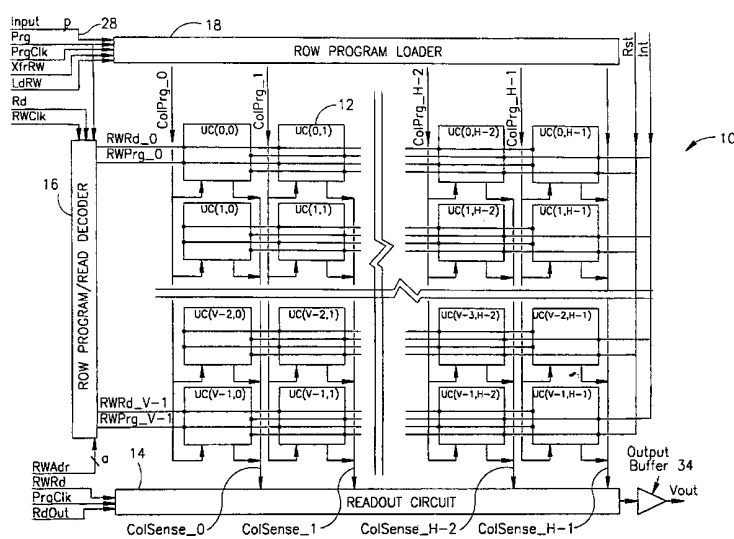
PCT

(10) International Publication Number
WO 01/10110 A2

- (51) International Patent Classification⁷: **H04N** (74) Agent: **EITAN, PEARL, LATZER & CO-HEN-ZEDEK**; Gav Yam Center 2, Shenkar Street 7, 46725 Herzlia (IL).
- (21) International Application Number: **PCT/IL00/00438**
- (22) International Filing Date: **24 July 2000 (24.07.2000)** (81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (25) Filing Language: **English**
- (26) Publication Language: **English**
- (30) Priority Data:
60/145,960 29 July 1999 (29.07.1999) US
09/516,168 29 February 2000 (29.02.2000) US
- (71) Applicant (for all designated States except US): **VISION - SCIENCES INC.** [US/US]; 40 Ramland Road South, Orangeburg, NY 10962 (US).
- (84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- (72) Inventor; and
(75) Inventor/Applicant (for US only): **STARK, Moshe** [IL/IL]; Harimonim Street 9, 40500 Even Yehuda (IL).
- Published:
— Without international search report and to be republished upon receipt of that report.

[Continued on next page]

(54) Title: **IMAGE-SENSOR ARCHITECTURE FOR PER-PIXEL CHARGE-INTEGRATION CONTROL**



(57) Abstract: A sensor array that produces a captured image at the end of a time frame. The array includes a plurality of unit cells which sense the image with multiple within-frame charge-integrations, and control means which separately controls each of the unit cells. The unit cells are programmable multiple charge-integration unit cells with modes of photocurrent integration and non-integration. The control means includes means for separately controlling multiple charge-integrations in a single frame capture of each unit cell, independently of the charge-integrations of the other unit cells.

WO 01/10110 A2



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

IMAGE-SENSOR ARCHITECTURE FOR PER-PIXEL CHARGE-INTEGRATION CONTROL

FIELD OF THE INVENTION

5 The present invention relates to image sensor array architecture generally and, in particular, to logic control thereof.

BACKGROUND OF THE INVENTION

10 Image sensors are generally comprised of an array of sensing unit cells, wherein each unit cell comprises a pixel which is exposed to light, and produces an electrical response representative thereof. Hereinbelow are defined the basic terms used in relation to image-sensor technology and several known-in-the-art methods for the same.

15 The minimal signal that can be detected by an image sensor is defined as the minimal incident light intensity on the pixel that results in a recognizable, meaningful response signal above the noise level. Signals with light intensity below the noise level are considered to act in the image sensor's cutoff region.

20 The maximum signal that can be detected by an image sensor is defined as the maximal incident light intensity on an image sensor's pixel that results in a recognizable non-saturated response. Signals with light sensitivity above this level are considered to be in the saturation range.

 The region between the cutoff region and the saturation range is defined as the image sensor's sensitivity range. Light signals with intensity in the image sensor's sensitivity range yield a response signal that corresponds to the incoming light intensity

25 The resolution and the minimum sensitivity determine the noise floor. Dynamic range (DR) performance is described in terms of the ratio between the highest intensity and the lowest intensity range limits. An image sensor's dynamic range is described in three equivalent ways.

 In the first way, the dynamic range is described as the ratio:

30 (1) $DR^1_L \equiv 10^n : 1$

where DR_L^1 is the image sensor's dynamic-range performance and n is a positive number, normally rounded to an integer.

Hence, an image sensor with a dynamic range of $10^3 : 1$ can capture signals that are up to thousand times larger than its minimum signal.

5 The second way of describing the dynamic range is in a logarithmic fashion, where:

$$(2) \quad DR_L^2 \equiv 20 \cdot \log_{10} DR_L^1$$

The third way often used to describe the image sensor's dynamic range is by the number of bits required to describe the dynamic range in a binary number fashion. This number of bits is directly related to the dynamic range by the following formula,

$$(3) \quad N_b \equiv \text{Intg} (\log_2 DR_L^1 + 1),$$

where; N_b is the number of bits, and Intg is a function that extracts the integer part of its argument.

15 Ideally, the most desirable image sensor is one that imitates the human eye's performance and captures scenes with comparable performance to the human eye's retina. However, while the human eye's retina provides a dynamic range of $10^8 : 1$, commercially available image sensor's "silicon retinas" provide a dynamic range that is typically only $10^3 : 1$. Thus, in comparison to the human eye's dynamic-range performance, the silicon retina performs quite poorly.

Dynamic range is a central issue in image-sensor design research. The basis for the research is the understanding of the workings of the human eye's retina. The superb performance of the human eye's retina results from the fact that each retina's photoreceptor locally adjusts its sensitivity to the intensity of the incident light. Although the individual photoreceptors of the human eye's retina each have a dynamic range of less than $10^3 : 1$, the overall retina's performance is much better due to photoreceptor's capability to locally adjust its "quiescent point". Shifting the point of operation means that the photoreceptor, when exposed to a high intensity of light, reduces its sensitivity while, when exposed to a low light intensity, it increases its sensitivity.

The research into improvement of the artificial retina's dynamic range is intensive, and today takes one of the following forms:

- **Logarithmic Sensors:** This type of sensor logarithmically compresses the dynamic range. The logarithmic compression is done by a logarithmically behaving photosensor, or a logarithmically responding circuit to an input photocurrent.

5 These circuits however, are quite sensitive to the manufacturing process, and slight variations in the manufacturing process may result in varying pixel-response sensitivities. Even adjacent pixels may significantly vary in their response sensitivity. This variance expresses itself in Fixed Pattern Noise (FPN), or in other ways that result in a poor quality image.

10

- **Multiple Exposure Sensors:** Several images at different exposures (charge-integration periods) are acquired and then combined into a single image. Typically, the combination of several different-exposure images is done on the image sensor's video output.

 Due to image acquisition time, and to computing-
15 intensive/time-consuming image-combination constraints, this method is typically restricted to the acquisition and processing of two images. The drawback of this method is that if the pair of acquired images differ substantially in their exposure times, the outcome can be image-color artifacts and edge artifacts.

20

- **Autonomous/Per-Pixel Controlled-Exposure Time Sensors:** For this approach, each pixel's exposure time is independently controlled and locally adjusted to the incident light's intensity. Efficient implementation of this method should yield the best results. Two noticeable attempts in this direction have been reported so far.

25 One reported method is based upon a unit cell that incorporates a static Set-Reset flip-flop. Resetting each pixel at a programmable point in time triggers the charge accumulation and thus controls the charge-integration time. Unfortunately the result is a large unit-cell area, a low fill factor, or both. Therefore, this unit cell is not suitable for small-pixel/high-resolution image
30 sensors.

 Furthermore, the image sensor's dynamic range is highly dependent on the column scan rate. Column-scan rate is limited by the programming rate of

each column. For instance, if the column-scan rate is to be performed at the pixel clock rate (which is by itself problematic and contributes to noise), each column program must be loaded in a single pixel time. This can be done only with a very wide and fast bus that loads the row control register.

5 S. G. Chen and J. P. Lee discuss the use of flip-flops in their article "Adaptive Sensitivity CCD Image Sensor: Charge-Coupled Device and Solid-State Optical Sensors V", *Proc. SPIE*, Vol. 2415, pp. 303-309, 315. However, the authors do not explain therein the details of implementation. The article describes an implementation limited to three different exposure intervals
10 with a ratio of 64: 8: 1. For many scenes, these steps are generally too crude to eliminate the quantization artifacts.

 A second reported method is the Independent Pixel Reset (IPR) method. This method requires an independent reset of integration capacitor charge in every unit cell. Each unit cell is individually reset through a row reset
15 control, and a column reset control circuits. This method results in an area-efficient solution.

 However, the method is limited in its scope since unit cells with the same exposure time are not reset simultaneously. Therefore the unit cells at the upper left corner of the image sensor array are reset significantly sooner than
20 the unit cells at the bottom right corner. For instance, if the array is 768×483 lines, and the reset is performed with a 67-nsec pixel clock, the time difference between the upper left corner reset and the lower right corner reset is 24.86 milliseconds, which is unacceptable for commercial video applications. Hence, the method is not suitable for large arrays or for real-time 30 frames per second
25 video rate.

 Additionally, the exposure time of the sensor is limited to a few values, such as 1, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, and $\frac{1}{16}$ of the maximum integration time, which is unsatisfactory for real-time high quality imaging. Hence, the IPR technique is currently unsuitable for most commercial applications.

SUMMARY

It is an objective to provide a novel, improved CMOS image sensor architecture that facilitates improved performance and dynamic-range scene capture.

5 There is therefore provided in accordance with a preferred embodiment of the present invention, a sensor array that produces a captured image at the end of a time frame. The array includes a plurality of unit cells which sense the image with multiple within-frame charge-integrations, and control means which separately controls each of the unit cells. The unit cells are programmable
10 multiple charge-integration unit cells with modes of photocurrent integration and non-integration.

Each unit cell includes a photosensor, a charge-storage device for accumulating charge transfer from the photosensor, and a programmable memory unit for storing a charge-integration state of the unit cell.

15 The control means includes means for separately controlling multiple charge-integrations in a single frame capture of each unit cell, independently of the charge-integrations of the other unit cells.

The control means includes means for providing N charge-integration sub-periods within a single frame capture, where N is equal to or greater than 1.

20 Within each single frame capture, the control means generally simultaneously individually integrates the charge of each unit cell within a group of cells. Generally, the control means includes a row-select line and a column-select line. The row-select line carries a multiplicity of first signals to the unit cells and the column-select line carries a multiplicity of second signals, such
25 as program and sense signals, to the unit cells.

Furthermore, the control means also programs the unit cells. A group of cells, such as a line of unit cells, is generally simultaneously programmed, preferably in a sequential order. The means for programming may also include means for sequentially programming one or more lines of the plurality of unit
30 cells.

The control means includes means for defining the charge-integration sub-periods where, alternatively, the charge-integration sub-periods are of

various time lengths. The control means also includes means for providing fine time resolution in clock-time units and means for providing wide dynamic range of charge-integration steps. Generally, the wide dynamic range is in the range of 2^N-1 unit steps of integration times.

5 There is therefore further provided, in accordance with a preferred embodiment of the present invention, a method for sensing an image with a plurality of unit cells. The method includes the steps of individually accessing each of the unit cells and controlling charge-integration of each unit cell, independently of the charge-integration of other of the plurality of unit cells.

10 The method further includes the steps of determining the charge-integration time for each unit cell and programming each unit cell according to the determined charge-integration time.

 There is therefore further provided in accordance with a preferred embodiment of the present invention, a method for improving the intra-scene
15 dynamic range of an image-sensor array of multiple unit cells. The method includes the steps of individually accessing and individually controlling each unit cell.

 The method also includes the step of individually controlling the charge-integration time of each of the unit cells, including individually
20 programming such. The step of individually programming includes programming each unit cell according to a pre-determined charge-integration time and programming each unit cell with multiple charge-integration sub-periods.

 There is therefore further provided, in accordance with a preferred embodiment of the present invention, a programmable image sensor including a
25 multiplicity of unit cells for capturing an image. The sensor includes a first plurality P of input lines for carrying data, a second plurality H of columns which are connected to the cells, wherein P is equal or smaller than H, and a controller for receiving the data and selectively distributing the data to the columns, to program N times the array, within a single frame of motion video.

30 Generally, the data is programming data and includes charge-integration/non-integration state data for each of the plurality of unit cells.

Preferably, each of the plurality of programmable unit cells is individually controlled.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be understood and appreciated more fully from the following detailed description taken in conjunction with the appended drawings in which:

5 Fig. 1 is a schematic illustration of an image-sensor array architecture for implementation with non-interlaced video, constructed and operative in accordance with a preferred embodiment of the present invention;

 Fig 2 is a schematic illustration of an image-sensor unit cell that provides multiple integration sub-periods and is used with the architecture illustrated in Fig.
10 1;

 Fig. 3 is a schematic illustration of a row-program loader used in the architecture illustrated in Fig. 1, and constructed and operative in accordance with a preferred embodiment of the present invention;

 Fig. 4A and 4B are timing diagrams of single row programming when
15 implemented with the architecture illustrated in Fig. 1;

 Fig. 5 is a timing diagram of a programming sequence implemented with the architecture illustrated in Fig. 1;

 Fig. 6 is a timing diagram of a programming/integration interval operative with the architecture illustrated in Fig. 1;

20 Fig. 7 is a schematic illustration of an image-sensor array architecture for implementation with interlaced video, constructed and operative in accordance with an alternative preferred embodiment of the present invention;

 Fig 8 is a schematic illustration of an alternative image sensor unit that provides multiple integration sub-periods and is used with the architecture
25 illustrated in Fig. 7;

 Fig. 9 is a schematic illustration of a row-program loader used in the architecture illustrated in Fig. 7, and constructed and operative in accordance with an alternative preferred embodiment of the present invention;

 Fig. 10A and 10B are timing diagrams of single-row programming
30 implemented with the architecture of Fig. 7.;

 Fig. 11 is a timing diagram of even-row programming implemented with the architecture of Fig. 7.; and

Fig. 12 is a timing diagram of an interlaced readout, program and integration cycle implemented with the architecture of Fig. 7.

DETAILED DESCRIPTION OF THE PRESENT INVENTION

Herein is detailed an innovative CMOS image sensor architecture and method that facilitates independent control of the charge-integration time of each pixel. In a preferred embodiment, autonomous per-pixel exposure control is accomplished via direct and independent control of the pixel's charge-integration time.

The present invention describes two novel image-sensor architectures, for utilization in both interlaced and non-interlaced video. Preferably, the image-sensor architectures incorporate a multiplicity of image-sensor unit cells that are capable of multiple charge-integration sub-periods, and preferably, the architectures provide for individual charge-integration time per pixel.

Implementation of the present invention facilitates programming the charge integration time in numerous, various-timed, small steps. As a result, hundreds, or even thousands, of exposure-time values are available. This is important for an effective implementation of dynamic range compression, and elimination of quantization noise and image artifacts. Hence, exposure times in the range of $2^{10} : 1$ in unit steps are feasible for motion video. This is instrumental for accomplishing wide dynamic-range scene capture roughly comparable in performance to the human eye's retina.

Furthermore, via novel use of a row-program loader, the present invention provides for simultaneous different charge-integration intervals for all the pixels. The present invention describes loading an entire row program, one row at a time, at video rate. Furthermore, this operation is done outside the image sensor's array.

Reference is now made to Fig. 1 an illustration of array 10, a novel and improved non-interlaced type CMOS image-sensor array architecture that comprises a plurality of unit cells 12 and implements autonomous, per-pixel, charge-integration control.

Reference is now made in parallel to Fig. 2, an illustration of unit cell 12, the basic building block of array 10. Unit cell 12 captures light and produces electrical signals associated therewith. Preferably, unit cell 12 is capable of multiple-integration sub-periods, as is the unit cell described in patent

application PCT/1L00/00129, CMOS Unit Cell with Autonomous/Per-Pixel Charge Integration Time Control Circuit, filed on March 2, 2000. Application PCT/1L00/00129 is co-pending to the same assignees as the invention presented herein, and is incorporated herein by reference.

5 For purposes of clarity herein, reference to a "unit cell" is synonymous with reference to a "pixel", and "integration" is synonymous with charge-integration.

 All the unit cells 12 in array 10 undergo a programming cycle, in a manner to be explained in detail hereinbelow, wherein they are preprogrammed
10 for a predetermined integration interval. Each such programming cycle is followed by the predetermined integration interval. Each such predetermined integration cycle may comprise a series of integration sub-periods and/or a series of non-integration sub-periods. At the completion of the integration interval the unit cell 12 is read out. The procedure of reset, programming cycle,
15 integration cycle, and readout is repeated. An integration interval is defined as the period of time from reset to readout.

 During the integration sub-periods the preprogrammed unit cell 12 injects charge into an integration capacitor which resides in unit cell 12. During the non-integration sub-periods, the integration capacitor stores the collected
20 charge, but the preprogrammed unit cell 12 does not inject additional charge into the capacitor. As explained in PCT/1L00/00129, the integration and non-integration sub-periods may be multiple and alternate between integration and non-integration, depending on the predetermined programmed cycle.

 During the integration interval, all the unit cells 12 that are primarily
25 enabled for charge integration are "exposed" to the light intensity. It is noted that although the integration interval is the same for all unit cells 12, the charge collected during this period differs from unit cell 12 to unit cell 12. The charge accumulated per unit cell 12 is proportional to the effective charge-integration time (i.e. the summation of the various integration sub-periods for each pixel)
30 and to the local intensity of the incident light.

Unit cell 12 retains the so-far accumulated charge on its integration capacitor, as long it is not actively discharged. In such a manner, unit cell 12 functions as an analog memory element which retains analog signals.

Hence, the total integration time and total accumulated charge of each unit cell 12 is dependent on its associated sequence of programmed states of integration/non-integration, such that each sub-period is subject to a different charge integration. Based upon PCT/1L00/00129, and for the special case where,

$$(4) \quad T_m = 2^m \cdot T_0, \quad k = 0, 1, \dots, q-2, q-1,$$

10 T_0 is the basic time unit,

T_m is the Int signal staying "high" in the i -th integration sub-period,

q is the number of program/integration sub-periods in the sequence,

T is the total charge integration time,

then,

$$15 \quad (5) \quad T = T_0 \cdot \sum_{m=0}^{q-1} u_m \cdot 2^m \quad \text{or}$$

and

$$(6) \quad T = T_0 \cdot (u_{q-1} u_{q-2} \dots u_2 u_1 u_0)_2$$

Therefore the integration interval is programmable through the binary number $(u_{q-1} u_{q-2} \dots u_2 u_1 u_0)_2$.

20 When $(u_{q-1} u_{q-2} \dots u_2 u_1 u_0)_2$ is $(0 \ 0 \dots 0 \ 1)_2$ T_i (min) = T_0 is the shortest possible integration time.

When $(u_{q-1} u_{q-2} \dots u_2 u_1 u_0)_2$ is $(1 \ 1 \dots 1 \ 1)_2$ T results in the longest possible integration time, which is,

$$(7) \quad T_{\max} = T_0 \cdot (2^q - 1),$$

25 where,

$T_{\max}$ is the maximum integration time,

q is the number of program/integrate sub-periods,

T_0 is the basic charge integration time unit.

30 $(u_{q-1} u_{q-2} \dots u_2 u_1 u_0)_2$ can be any integer number in between the minimum and the maximum values, stepped in T_0 units. Therefore, the

integration time can be any value starting with T_0 , $2T_0$, $3T_0, \dots$, in T_0 steps, and up to $T_{\max}$.

The collected charge or the corresponding voltage is proportional to the total integration time, and to the photocurrent magnitude. The photocurrent itself
 5 is proportional to the intensity of the incident light.

This is summarized in the following two formulas:

$$(8) \quad Q^a = I_{ph} \cdot T,$$

$$(9) \quad V^c = I_{ph}/C_I \cdot T,$$

where,

10 I_{ph} is the photocurrent, T is the total integration time,
 C_I is the Integration Capacitor's capacitance,
 Q^a is the accumulated charge on the Integration Capacitor, and
 V^c is the voltage across the Integration Capacitor.

The implementation of equations (4) – (9) will be explained in detail
 15 hereinbelow in connection with Figs. 6 and 12, respectively.

Referring briefly to Fig. 1, array 10 comprises an array of lines, arranged in rows along the x-axis and columns along the y-axis. Unit cells 12 are located at the x, y nodes of the lines, and connected thereto. The lines carry electrical signals that control the integration interval of each unit cell 12. For purposes of
 20 identification, each unit cell 12 is generally designated by its location on the x, y axis. As such, the unit cell 12 located at ($y = 0$, $x = 0$) is identified as unit cell 12 (0,0), and the unit cell located at ($y = V - 1$, $x = H - 1$) is identified as unit cell 12 ($V - 1$, $H - 1$).

Array 10 additionally comprises a multiplicity of circuits located on the
 25 horizontal and vertical axis, respectively. In one preferred embodiment, a readout circuit 14 and a row program loader 18 are located along the horizontal or y-axis, and a row program/read decoder 16 is located along the vertical or x-axis. It is noted that the usage of the terms vertical and horizontal are for purposes of clarity, and alternative locations of these circuits are included within
 30 the scope of the present invention.

In a preferred embodiment, the array of lines includes lines generally designated row-read (RwRd) lines, row-program (RwPrg) lines, column sense

(ColSense) lines, column program (ColPrg) lines, reset (Rst) lines, and integration (Int) lines. Each line carries an associated signal, i.e. the RwRd line carries a RwRd signal and so on.

The location designation of each line is represented by its respective
5 location on the x or y axis and is generally designated as such, i.e.: the ColPrg line located at $x = 0$ is designated as ColPrg_0 line, and the RwPrg line located at $y = i$ is designated as RwPrg_i line.

Int lines: The Int (i.e. integration) line is one line that splits into multiple lines and is connected to all unit cells 12. The Int line carries the Int signal. All
10 unit cells 12 are subject to integration controlled by the Int signal. Preferably, when the Int signal is high, charge integration takes place. During the integration sub period (when the Int signal is high), those associated unit cells 12 that have been programmed to integrate, do so.

It is noted that even if a specific unit cell 12 is programmed to integrate
15 during a specific sub-period, integration will take place only if the Int signal is high. Conversely, even if the Int signal is high, if a specific unit cell 12 is programmed not to integrate, then no integration will take place.

Rst lines: The Rst (i.e. reset) line is one line that splits into multiple lines and is connected to all unit cells 12. The Rst lines carry a Rst signal which
20 drives all the unit cells 12 in array 10 to reset simultaneously. The Rst signal activates a transistor that resides in unit cells 12 that drains the residual charge which remained after the previous readout, thereby preconditioning the integration capacitors for the next integration interval.

It is noted that, alternatively, reset does not always need to be
25 performed explicitly, but can be implicitly performed as part of the readout phase, in which case, the Rst line and the corresponding transistor are alternatively eliminated.

RwRd lines: In a preferred embodiment, the RwRd (i.e. row-read) lines are connected to the row program/read decoder 16, and the location
30 designations of the RwRd lines range from RwRd_0 to RwRd_V-1. Each RwRd line is connected to the associated unit cells 12 on its respective row. The RwRd lines carry a row read (RwRd) signal that controls the readouts of the

associated unit cells. When the $RwRd_i$ signal is high and all the other $RwRd$ signals are low, the content of the unit cells 12 (i,j) is read. Typically, unit cells 12 are readout after the completion of the charge integration cycle.

RwPrg lines: In a preferred embodiment, the $RwPrg$ (i.e. row-program) lines are connected to the row program/read decoder 16, and the location designations of the $RwRd$ lines range from $RwPrg_0$ to $RwPrg_V-1$. Each $RwPrg$ line is connected to all the associated unit cells 12 on its respective row. The $RwPrg$ lines carry a row program ($RwPrg$) signal that controls the programming of the associated unit cells 12. The programming of the unit cells 12 determines the charge integration sequence for each unit cell 12. Preferably, programming is done in a row-by-row fashion, i.e., when the $RwPrg_i$ signal is high and all other $RwRd$ signals are low, the unit cells 12 in row i are programmed.

It is noted that in a preferred embodiment the programming of unit cells 12 is performed before the readout starts, or off-line. Thus, the time-consuming task of programming does not affect the integration or read-out timing of array 10.

It is noted that the $RwRd$ signal and the $RwPrg$ signal function in tandem with each other. When the $RwRd$ signal is high, the $RwPrg$ signal is low, and visa versa.

RdEn and PrgEn signals: The $RwRd$ signal activates a $RdEn$ (i.e. read-enable) signal that enables the readout of the accumulated charge (current/voltage) of the associated unit cell 12. Similarly, the $RwPrg$ signal activates a $PrgEn$ (i.e. program-enable) that enables the programming of the associated unit cell 12.

ColSense and ColPrg lines: In a preferred embodiment, the $ColSense$ (i.e. column-sense) and $ColPrg$ (i.e. column-program) lines run between row program loader 18 and readout circuit 14. In one preferred embodiment, the $ColSense$ lines and $ColPrg$ lines are alternatively active; when the $ColSense$ line is active - the $ColPrg$ line is inactive, and visa versa. As such, the same physical line is usable for both functions. Alternatively, the $ColSense$ lines and the $ColPrg$ lines are separate physical lines.

The location designations of the ColSense/ColPrg lines range from ColSense_0/ColPrg_0 to ColSense_H-1/ColPrg_H-1. Each ColSense/ColPrg line is connected to all the associated unit cells 12 on its respective column.

In a preferred embodiment, the ColSense/ColPrg lines are common for programming and reading, and carry a ColSense/ColPrg signal. The ColSense/ColPrg signal is a multiplexed input/output signal that alternatively acts as an input signal during the charge integration interval, and as an output signal during the read cycle.

During the programming sequence, the ColSense/ColPrg lines serve as ColPrg_0/ColPrg_V-1 lines and provide the ColPrg signal, which is actively driven by row program loader 18. The value of the ColPrg signal determines whether the unit cell 12 will integrate or not integrate the photocurrent charge during the next integration sub-period.

When the ColPrg signal is low, unit cell 12 is preprogrammed to disable charge integration. Conversely, when the ColPrg signal is high, unit cell 12 is preprogrammed to enable charge integration. Alternatively, unit cell 12 can be programmed to respond to the opposite polarity of the ColPrg signal.

During the read cycle, the ColSense/ColPrg lines serve as ColSense_0/ColSense_H-1 lines and provide the ColSense signal. The outputs of row program loader 18 are tristated, and readout from the associated unit cells 12 is enabled, providing readout signals to readout circuit 14.

Row program/read decoder 16: Decoder 16 selects one or more of the RwRd lines and activates the RwRd signal, which enables readout of the charge accumulated in the associated unit cell 12. In a preferred embodiment, the readout phase of the selected unit cell 12 occurs when the RwRd signal is high, and simultaneously the RwPrg signal is low and no row programming is occurring.

In order to designate the appropriate RwRd line to be selected for readout, such as RwRd_z line, row program/read decoder 16 activates a predetermined combination of line address (RwAdr) input signals. The combination of RwAdr signals creates a binary combination representative of the number "z".

Conversely, during the programming phase of array 10, the RwRd lines are inactive, and the RwPrg lines are active, thereby allowing programming of the unit cells 12 in array 10, row-by-row. In a manner similar to RwRd activities, a predetermined combination of RwAdr input signals is activated for designation
 5 of the appropriate row for programming.

Selection of the appropriate RwAdr signals is calculated as follows: Let "a" be the number of RwAdr signals, then,

$$(10) \quad a \geq \log_2 V$$

$$(11) \quad z = (RwAdr_{a-1}, RwAdr_{a-2}, \dots, RwAdr_1, RwAdr_0)_2$$

10 (a) Selection of one of the combinations for which $z \geq V-1$ results in the deselection of all the RwRd lines and RwPrg lines.

(b) When all the RwRd lines are deselected and the RwRd signal is low, row program/read decoder 16 selects one of the RwPrg lines. Thus, for a particular RwPrg_z line, where $z = (RwAdr_{a-1}, RwAdr_{a-2}, \dots, RwAdr_1, RwAdr_0)_2$, all the unit cells 12 (z,j) on the RwPrg_z line are selected and are
 15 simultaneously programmed by the ColPrg₀/ColPrg_{H-1} lines connected thereto.

In a preferred embodiment a dynamic type of row program/read decoder 16 is suggested, and as such, a row clock (RwClk) signal is used for the internal
 20 decoder precharge. The precharge occurs before every row selection. Internal precharge results in the deselection of all RwRd lines and RwPrg lines by pulling all the lines to low. It is noted that other selection/deselection timing mechanisms are feasible and within the scope of the present invention.

Readout Circuit 14: Readout circuit 14 enables the readout of collected
 25 charge from each unit cell 12 in array 10. Readout circuit 14 typically comprises one sense amplifier (not shown) per each ColSense line, or column. The sense amplifiers sense the electrical signal, (either charge, voltage, or current) from the read-out unit cells 12, and convert the sensed signal into a more robust video signal.

30 The RwRd signal enables readout on a row by row basis, thus enabling the sense amplifiers to readout the unit cells 12 row by row. The output from each sense amplifier is then time multiplexed to the output of the readout circuit.

A pixel clock (PrgClk) signal, in coordination with a read out (RdOut) signal, is received in readout circuit 14, thus driving readout and orchestrating the readout timing of the unit cells 12. The signal representative of each of the collected charges is converted to video signals V_{out} that are sequentially output through an output buffer 34.

It is noted that there are many implementations possible for readout circuit 14. For instance, the video output could be demultiplexed, and readout simultaneously via several video outputs to speed-up the pixel video readout process.

In alternative preferred embodiments, different readout architectures and different output sequences are implementable. For instance, to speed up the readout, several pixels can be output at the same pixel period.

Row program loader 18: Reference is now made to Fig. 3, an illustration of a preferred embodiment of row-program loader 18. Row-program loader 18 facilitates programming each of the unit cells 12 to an integrate/non-integrate state. Preferably row-program loader 18 is loaded from a program memory, or from a charge-integration program generator, not shown. This memory or charge-integration program generator is either part of array 10 or resides outside of the architecture and is input to the program loader 18.

In a preferable embodiment, a single-row program is loaded into the internal memory of row-program loader 18 via a plurality of external input lines 28. The input pin of each input line 28 is connected to one of a plurality of shift register inputs 30, which are used to shift the row program in. Each shift register 30 comprises k (or r) stages 31. Preferably each shift stage 31 is a D type flip-flop, where in Fig. 3, an input to stage 31 is generally designated as D, an output is generally designated as Q, and a clock input is generally designated as CK.

It is noted that the total number of stages 31 corresponds to the total number of ColPrg lines in array 10.

As an example, array 10 comprises p shift registers 30. Thus, if:

H is the number of ColPrg lines in the array (and also the total number of stages 31),

p is the number of input lines 28 to the row program loader 18 (and also the number of shift registers 30), and if

p-1 shift registers 30 have k stages 31 and

one (1) register 30 has r stages 31, where $r \leq k$, then;

$$(12) \quad H = [(p-1) \cdot k] + (1 \cdot r) = (p-1) \cdot k + r$$

Example 1: In one preferred implementation of row-program loader 18, there are

H = 768 columns in array 10, and

p = 16 input lines 28, each connected to an associated shift register 30.

Each of the 16 shift registers 30 contains

k = 48 stages 31. Thus,

$$768 = 16 \cdot 48$$

It is noted that in this example, $r = 0$, thus $H = (p \cdot k)$.

Example 2: In an alternative preferred implementation of row-program loader 18;

H = 756 columns in array 10, and

p = 16 input line 28.

The p input lines 28 connect to

15 shift registers 30 which contain k = 48 stages 31, and

one shift register 30 which contains r = 36 stages 31. Thus,

$$756 = [(16-1) \cdot 48] + (1 \cdot 36) = (15 \cdot 48) + 36$$

In order to speed up the row-program loading, all the p shift registers 30 are simultaneously loaded by the p input lines 28. In a preferred embodiment of the present invention, input line 28 loads data into shift register 30 from the least significant bit to the most significant bit. Thus, if the programming data is input with the least significant data bit first, and in order, with the most significant data bit last, then after k shifts, the data resides in the right order (left to right), inside the shift register 30.

However, the exception to the rule is the last shift register 30, which has r stages 31, where r is less than or equal to k. Therefore, to synchronize the operation with other p-1 shift registers 30, the data for the last shift register 30

must start with k-r garbage/don't care information bits followed by r information bits.

The time it takes to load the entire line T_{LdRw} is.

$$(13) \quad T_{LdRw} = k \cdot T_p$$

or

$$(14) \quad T_{LdRw} = \cdot \text{Intg}(H/p + 1) \cdot T_p ,$$

Where Intg function is the integer part of the function argument,

Example 3: In this example of row program loader 18;

H = 768 columns in array 10, and

10 p = 32 input lines 28, each connected to an associated shift registers 30, each shift registers 30 containing k = 24-stages 31.

For a 57.272 MHz programming clock (four times the frequency of the pixel clock, i.e.: $4 \cdot 14.318 \text{ MHz} = 57.272 \text{ MHz}$), T_p = (the clock time period) is 17.46 nsec,

15 thus T_{LdRw} , the time to load an entire line is 419.04 nsec.

$$419.04 \text{ nsec} = 24 \cdot 17.46,$$

or

$$419.04 \text{ nsec} = [1 \cdot (\frac{768}{32})] \cdot 17.46$$

20 At the shift-in of the last bit of the row program, the entire program is transferred into a line register LR, preferably comprises of a series of D type flip-flops.

25 The data, or program, stored in the line register LR is transferred via a row load (LdRw) line to a set of H tristate buffers 32. Each tristate buffer 32 drives an associated ColPrg line, enabling programming of the associated column.

30 As noted above, the value of the ColPrg signal determines whether the unit cell 12 will integrate or not integrate the photocurrent charge during the integration/non-integration sub-period. Hence, the data loaded by the buffers 32 to the ColPrg line is a series of values (0s and 1s) that determine the next integration/non-integration sub-period, of the associated unit cells 12.

Once the data is transferred to the line register LR and programming of the relevant columns commences, the shift registers 31 are free to receive data for the program load of the next line. Hence, the programming of ColPrg_i line occurs simultaneously with the loading of the ColPrg_{i+1} line after it.

5 For embodiments where there are V rows, the time to load the entire array 10 is expressed by the following equation:

$$(15) \quad T_{\text{Prg}} = V \cdot \text{Intg}(H/p + 1) \cdot T_p,$$

Example 4: For the case described in Example 3, and for an array 10 (array) with:

10 $V = 483$ rows, and therefore

$$T_{\text{Prg}} = 202.39632 \mu\text{sec} = 483 \cdot \left[1 \cdot \left(\frac{768}{32}\right)\right] \cdot 17.46$$

Reference is now made in parallel to Figs. 4A and 4B, timing diagrams illustrating a single line of programming as performed by row-program loader 18. It is noted that the signals depicted in Figs. 4A and 4B are received from, or act upon, the hardware depicted in Fig. 3. As such, it is noted that the reference
15 names are similar, i.e. XfrRw (i.e. transfer-row) line sends a XfrRw signal, and input lines 28 send input data signals.

The top line of Fig. 4A depicts the program clock (PrgClk) pulse. With every rise of the PrgClk pulse, a time period T_p commences. Hence, time period
20 T_p extends from PrgClk pulse rise to PrgClk pulse rise.

Each time the PrgClk signal goes low, new data (Input₀, Input₁, Input_{p-2} and Input_{p-1}) is presented to inputs 28. Each time the PrgClk goes high, the data into (Input₀, Input₁, Input_{p-2} and Input_{p-1}) is loaded into the stages 31. The input lines 28 consecutively load the data (input signals),
25 shifting it from stage 31 to stage 31, until the entire array has been loaded.

In order to use time efficiently, and as explained in connection with Fig. 3, each input line 28 loads an associated shift registers 30 with k stages 31. Hence, after k PrgClk signals, each input line 28 has loaded k stages 31. It is additionally noted that for purposes of clarity, Fig. 4A depicts only four such
30 input signals, associated with four input lines 28. However, it is obvious to those skilled in the art that the present invention entails the appropriate number of

input signals for the relevant number of input lines 28 which feeds the row loader 18.

After all the input signals have been loaded into the stages 31, a transfer row (XfrRw) signal (Fig. 4B) releases connection from the row which has just completed programming and connects to the next row designated for programming. The XfrRw signal is preferably an externally controlled signal.

The row load (LdRw) line then releases a LdRw signal that enables the buffers 32 during programming. As depicted in Fig. 4B, if the XfrRw signal releases row $i - 1$ for loading, then the LdRw signal loads row $i-1$.

Depicted simultaneously with the LdRw signal, is the ColPrg_j signal, the RwAdr signal, and a RwPrg signal. The ColPrg signal is the data sent to the specific row via the buffers 32. The ColPrg signal contains the data specifying the program instructions for the associated column, i.e. the ColPrg_j signal contains the programming instructions for the column j .

The RwAdr signal contains the address of the row to be loaded, and causes the data to be loaded to the appropriate row. The RwPrg signal contains the data specifying the program instructions for appropriate associated row.

As is noted hereinabove, array 10 is set up as a matrix of columns and rows. Row loader 18 loads data for a selected row and programming information is sent for all the columns that intercept that row. Consequently, the unit cells 12 that are affected are the pixels of that selected row.

Reference is now made to Fig. 5, which details the timing diagram of one programming sequence of the entire array 10. The top line of Fig. 5 depicts a Prg signal, which remains high during a time period T_{prg} , which is a composite of multiple time periods T_{LdRW} , as defined in connection with Fig. 3 and equation 13. T_{prg} is the time period required for one programming sequence of the entire array 10.

A series of XfrRw signals goes high releasing a series of rows designated to be programmed. Simultaneously with the fall of each XfrRw signal, the associated LdRw signal goes high. Thus, when XfrRw signal releases row 0, the LdRw signal 0 goes high, thereby preparing row 0 to receive the line register data.

With each commencement of a LdRw signal, all the ColPrg signals and the RwAdr signal change states, performing their appropriate tasks as described in Fig. 4.

As depicted in Fig. 5, when the LdRw signal for a particular row goes high, only the RwPrg signal for the row associated with that particular row goes high. Thus, simultaneously with the rise of LdRw_0 signal, RwPrg_0 commences programming row 0, and simultaneous with the rise of LdRw_1 signal, RwPrg_1 commences programming row 1. Hence it is possible to see from Fig. 5 the progression of the row programming from row 0 to row V - 1.

It is noted that the Int signal (the bottom line in Fig. 5) is low during the programming sequence, thus disabling the charge integration. Thus, integration does not occur during the programming sequence, and programming is enabled to take place without interfering with the integration process.

Reference is now made to Fig. 6, a timing diagram of the programming and integration sequences of array 10. Array 10 is programmed and integrated q times, whereas each integration sub-period T_q is preceded by programming of the entire array 10.

As illustrated in Fig. 6, each programming sequence is followed by an integration sub-period. In a preferred embodiment of the present invention, the integration sub-period is halved in every cycle, as such, the second integration sub-time T_{q-2} is half the time of the first integration sub-period T_{q-1} .

Each unit cell 12 is subject to integration as governed by a unique set of u_m programming coefficients, defined by a coefficient vector $(u_{q-1} u_{q-2} \dots u_2 u_1 u_0)_2$. It is noted that the programming coefficients u_m define the state of integration or non-integration for each preprogrammed integration sub-period T_m , such that if u_m is high (1), integration takes place, and if u_m is low, integration does not take place.

Thus, the total integration time T is governed by the programming coefficients u_m , and by the basic time integration unit T_0 , as described by formula (6). Hence, each unit cell 12 is subject to a unique and autonomously defined charge integration, i.e.: charge integration time.

As illustrated in Fig. 6, when the Rst signal goes high the single frame programming/integration/readout commences. The Rst signal is immediately followed by the first array programming cycle.

During the first charge integration, the most significant programming coefficients $u_{q-1,i,j}$ are programmed into the respective unit cells 12. The first array programming is followed by charge integration over the entire array 10. For unit cells 12 with multi-charge integration sub-periods, as described in application PCT/1L00/00129, if $u_{q-1,i,j} = 1$ then unit cell 12 (i,j) is subject to charge integration for a period $T_{q-1} = 2^{q-1} \cdot T_0$. However, if $u_{q-1,i,j} = 0$, then unit cell 12 is not subject to integration during this charge integration step. This cycle, is followed by another, which starts with the programming of the entire array, this time with $u_{q-2,i,j}$ coefficients, and followed by another charge integration step, this time for a period $T_{q-2} = 2^{q-2} \cdot T_0$.

The sub-period charge integrations continue until the last charge integration, in which the array is programmed by coefficients $u_{0,i,j}$, and is subject to the shortest integration $T = T_0$. In each charge integration, additional charge is accumulated in the integration capacitor, and retained until the next charge integration. Effectively, the integration time for each pixel is different, and governed by the programming coefficients u_m as defined by formula (6).

It is noted that although herein the integration sub-period T_m varies from longest to shortest, other variations of timing schedules are also applicable within the scope of this invention.

The programming/integration interval is followed by the image sensor array readout. If,

T is the total integration time
 T_{Rd} is the image sensor readout time,
 T_{Prg} is the total programming time,
 T_{FR} is the time allocated to a single frame, and
 q is the number of program/ integration cycles,
 then,

$$(16) \quad T_{FR} = T + q \cdot T_{Prg} + T_{Rd}$$

Example 5: If the image sensor described in Examples 3 and 4, is subject to;

$q = 10$ program/integrate cycles.

$T_{Rd} = 8$ msec for image sensor readout.

5 $T = 23.31$ msec for integration.

T_{prg} from Example 4 is $202.39632 \mu\text{sec}$, and therefore,

For motion video the frame rate is 30 frames per second such that $T_{FR} = 33.333$ msec,

The time consumed by 10 program cycles is 2.023 msec,

10 $33.333 \text{ msec} = 23.31 \text{ msec} + 10 \cdot 202.39632 \mu\text{sec} + 8 \text{ msec}$

It is noted that since there are 10 integrations, 1023 different charge-integration values are possible, stepped up to 23.31 msecs.

It is noted from the above example that it is possible for each pixel to be individually and autonomously programmed with a wide range of charge-integration times, programmed in time unit steps. This is instrumental in the capture of wide dynamic range scenes and furthermore important for the elimination of quantization noise. The time consumed by the programming can be designed to become reasonably short and therefore to slightly affect the maximum integration time.

20 Another useful aspect of the present invention is a wide charge-integration time dynamic range (DR_T). If the charge-integration time dynamic range (DR_T) is defined as,

$$(16) \quad DR_T = 20 \cdot \log_{10} (T/T_0) ,$$

where T is the total integration time

25 and T_0 is the basic time integration unit;

then,

$$(17) \quad DR_T = 20 \cdot \log_{10} (2^q - 1)$$

For $q \gg 1$,

$$(18) \quad DR_T \approx 6.02 \cdot q$$

30 The integration-time dynamic range DR_T parameter is the one of the keys to accomplishing high dynamic ranges of the artificial retina. A wider program-load bus and/or a faster program clock enable more

program/integration cycles, and therefore a wider integration time dynamic range.

Consequently, when constructed and operated according to a preferred embodiment of the present invention, array 10 provides a multiplicity of programming and charge-integration sub-periods within a single integration interval (or equivalently within a single frame). The presented case of a circuit working at about 60 MHz program clock can be accomplished with the current state-of-the-art CMOS technologies. However, care must be taken to eliminate any possible effects of the program clock on the signal noise. Since, the programming is done via the row-program loader 18, which is outside array 10, the circuit must be carefully isolated, and uncoupled from the array. For instance, an appropriate guard ring around the row-program loader's 18 layout could help to accomplish this goal.

Interlaced image sensor

TV-format image sensors usually operate in an interlaced mode. In the interlaced mode, the frame output is divided into two fields. In the first field period the lines 0, 2, 4, ... are read out. In the second period lines 1, 3, 5, ... are read. The readout timing is compliant with the TV-format timing whether this is an NTSC, or a PAL format. For an interlaced image sensor, the program/integrate cycles are performed on one field, while readout is performed on the other.

Reference is now made to Figs. 7 and 8, which illustrate different aspects of an implementation of autonomous/per-pixel charge-integration control for an interlace-type CMOS image-sensor array 50.

Fig. 7 is a block diagram of image-sensor array 50. Fig. 8 depicts a unit cell 52, which is used in array 50. Elements similar to array 10 are similarly labeled and will not be described further.

In a preferred embodiment, image sensor array 50 has H ColPrg/ColSense columns, and V RwPrg/RwRd lines (rows), where V is assumed to be odd. Preferably, the odd ColPrg lines are programmed simultaneously with the readout of the even ColSense lines.

Hence, for this preferred implementation, unit cell 52 requires that the ColPrg lines be separated from the ColSense lines.

However, other aspects of unit cell 52 are similar to unit cell 12, and hence will not be described further.

5 Image sensor array 50 comprises two row-program/read decoders 16E and 16D. While one line decoder controls the readout, the other controls the programming, and vice versa.

Row-program/read decoder 16E controls the even read (RwRd) and program (RwPrg) lines. During a read cycle, decoder 16E generates even RwRd
10 lines, such as, RwRd_0, RwRd_2, RwRd_4, RwRd_6..... During the program/integrate cycles, decoder 16E generates even RwPrg lines, such as, RwPrg_0, RwPrg_2, RwPrg_4, RwPrg_6 and so on.

The operation of decoder 16E is controlled by a plurality of signals, a read-even (RdEven) signal and a program-even (PrgEven) signal. When the
15 RdEven signal is low and the PrgEven signal is high, the unit cells 52 on the even RwPrg lines are activated. When RdEven is high and PrgEven is low, the unit cells 52 on the even RwRd lines are activated. When RdEven, and PrgEven are both low, the RwRd and RwPrg lines from decoder 16E are all low, and therefore inactive.

20 Row-program/read decoder 16D controls the odd read, and program lines. During a read cycle, decoder 16D generates the odd read lines, such as, RwRd_1, RwRd_3, RwRd_5, RwRd_7.... During the program/integrate cycles, decoder 16D generates the odd program lines, such as RwPrg_1, RwPrg_3, RwPrg_5, RwPrg_7....

25 The operation of decoder 16D is controlled by a plurality of signals, a read-odd (RdOdd) signal and a program-odd (PrgOdd) signal. When RdOdd is low and PrgOdd is high, the unit cells 52 on the odd RwPrg lines are activated. When the RdOdd signal is high and the PrgOdd signal is low, the unit cells 52 on the odd RwRd lines are activated. When the RdOdd signal and the PrgOdd
30 signal are both low, the RwRd and RwPrg lines of decoder 16D are all low and therefore inactive.

In a preferred embodiment, the RdEven signal is in opposite polarity to the RdOdd signal, and the PrgEven signal is in opposite polarity to the PrgOdd signal. In this example, it is noted however, that all lines are the same polarity when all the RwRd lines and RwPrg lines are deactivated. Therefore, when the
5 decoder 16E generates even-program control signals, the decoder 16D generates odd-read control lines, and visa versa.

In order to conform with the present interlaced embodiment, readout circuit 14 alternately reads unit cells 52 on the even and odd numbered RwRd lines. Preferably, the unit cells 52 on the even numbered RwRd lines are read
10 first, and then the unit cells 52 on the odd numbered RwRd lines.

Since the program/integrates occur for the even unit cells 52 simultaneously with the read out of the odd unit cells 52, and vice versa, preferably Rst signals and Int signals are separate for even and odd fields.

Hence, an integrate-even (IntEven) signal activates integration of unit
15 cells 52 on the even Int lines. The IntEven signal is inactive during the readout of the unit cells 52 on the even RwRd lines.

Additionally, a reset-even (RstEven) signal activates reset of unit cells 52 on the even Rst lines. The Rst signal is inactive during the readout of the unit cells 52 on the even RwRd lines.

20 An integrate-odd (IntOdd) signal and a reset-odd (RstOdd) signal perform similar functions for the unit cells 52 on the odd lines.

Reference is now made to Fig. 9, an illustration of row-program loader 58, Fig. 10, a timing diagram depicting the single row-programming utilization of loader 58, and Figure 11, a timing diagram of even-field lines programming.
25 Elements similar to previous embodiments are similarly labeled and will not be described hereinbelow.

It is noted that the flow depicted in Fig. 9 is similar to that depicted in Fig. 3, and functions similar to that described hereinabove in reference to Fig. 3. It is however noted, that unit cell 52 has separate ColPrg lines and ColSense
30 lines, and hence tristates are not required and thus not depicted in Fig. 9.

It is additionally noted that the timing depicted in Figs. 10 and 11 is similar to that depicted in Figs. 4 and 5, respectively, and functions similar to that described hereinabove in reference to those figures.

It is however noted, that while Fig. 11 illustrates both a PrgEven signal and a PrgOdd signal, Fig. 5 depicts only a Prg signal. Nevertheless, while the PrgEven signal is high, the PrgOdd signal is low, and visa versa, and thus the appropriate high signal (PrgEven signal) functions similarly to that of the Prg signal as described in connection with Fig. 5, and is thus applicable to Fig. 11.

Although not illustrated in Fig. 11, while the unit cells 52 in RwPrg lines 0, 2, 4, ... are programmed, the unit cells in RwRd lines 1, 3, 5, ... are readout. Right after the programming of the even RwPrg line unit cells 52, the IntEven signal goes high, which result in charge integration initiation in the even-row unit cells 52. However, since the IntOdd stays low, no charge integration occurs in the odd-row unit cells.

The programming cycle for each of the T_{Prg} cycles, either even or odd, is halved, since only half of the rows are programmed. Such that,

$$(18) \quad T_{Prg} = \text{Intg}(V/2 + 1) \cdot \text{Intg}(H/p + 1) \cdot T_p$$

The programming cycle for the odd-row field is similar to the even-row field, although if the total number of lines is odd, it may be a bit shorter since there is one line less to program

Example 6: For the case described in Examples 3 and 4, but for an interlaced readout image sensor. Based upon formula (20) it takes 101.40768 μsec to program the array's even-row field.

Reference is now made to Fig. 12, a timing diagram of the programming and integration cycle of array 50 for interlaced readout. Note that the even and odd fields are programmed in tandem, one after the other.

The charge integration of the even-row field starts with RstEven, which discharges the integration capacitors inside the even-row unit cells 52. Then, the even-row field unit cells 52 are subject to q program/integrate cycles. These cycles are, in principle, similar to the ones described herein above for the non-interlace image sensor array 10, and will not be described further.

For this particular case, the even-field programming is performed in one field time, while the readout occurs in the other. Therefore,

$$(19) \quad T_{FL} = T + q \cdot T_{Prg} ,$$

where,

5 T_{FL} is the field period.

Example 7: Image sensor similar to the one described Example 3, and in Example 5, but of interlaced readout type is subject to 10 program/integrate cycles. The frame rate is 30 frames per second, and the field rate is 60 fields per second. For this case $T_{FL} = 16.666$ msec, the time consumed by 10 program
10 cycles is 1.014 msec, which leaves 15.652 msec for integration. Since there 10 cycles, there are 1023 different charge integrations possible, and stepped up to 15.652 msec.

Although the above invention describes apparatus wherein the active signal is high and the non-active signals are low, it will be apparent to those
15 skilled in the art that reverse polarity and components are applicable and included within the scope of the present invention.

The methods and apparatus disclosed herein have been described without reference to specific hardware or software. Rather, the methods and apparatus have been described in a manner sufficient to enable persons of
20 ordinary skill in the art to readily adapt commercially available hardware and software as may be needed to reduce any of the embodiments of the present invention to practice without undue experimentation and using conventional techniques.

CLAIMS

1. A sensor array which produces a captured image at the end of a time frame, said array comprising:
 - a plurality of unit cells for sensing said image with multiple within-frame
 - 5 charge-integration; and
 - control means for separately controlling each said unit cell.
2. An array according to claim 1, wherein said charge integration is capacitor discharge.
3. An array according to claim 1, wherein said unit cells are programmable
- 10 multiple charge-integration unit cells having photo-current integration and non-integration states.
4. An array according to claim 1, wherein said control means includes means for separately controlling multiple charge-integrations in a single frame capture of each said unit cell, independently of said charge-integrations of other said
- 15 cells.
5. An array according to claim 1, wherein said control means includes means, in each said single frame capture, for generally simultaneously individually integrating the charge of each said unit cell of a group of said cells.
6. An array according to claim 1, wherein said control means comprises;
 - 20 a row-select line for carrying a multiplicity of first signals to said unit cells; and
 - a column-select line for carrying a multiplicity of second signals to said unit cells, wherein said second signals comprise program and sense signals.
7. An array according to claim 1, wherein said control means includes means
- 25 for programming said unit cells.
8. An array according to claim 7, wherein said means for programming includes means for generally simultaneously programming a group of said cells.
9. An array according to claim 8, wherein said group of said cells is a line of said unit cells.
- 30 10. An array according to claim 9, wherein said means for programming includes means for sequentially programming said line of said unit cells.

11. An array according to claim 9, wherein said means for programming includes means for sequentially programming one or more said lines of said plurality of unit cells.
12. An array according to claim 4, wherein said control means includes
5 means for providing N charge-integration sub-periods within said single frame capture, where N is equal or greater than 1.
13. An array according to claim 12, wherein said control means includes means for defining said charge-integration sub-periods.
14. An array according to claim 12, wherein said charge-integration
10 sub-periods are of various time lengths.
15. An array according to claim 1, wherein said control means includes means for providing fine time resolution in clock-time units.
16. An array according to claim 1, wherein said control means includes means for providing wide dynamic range of charge-integration steps.
- 15 17. An array according to 16, wherein wide dynamic range is in the range of 2^N-1 unit steps of integration times.
18. An array according to claim 1, wherein each said unit cell includes:
a photosensor;
a charge storage device for accumulating charge from said photosensor; and
20 a programmable memory unit for storing a charge-integration state of said unit cell.
19. A method for sensing an image with a plurality of unit cells, said method comprising the steps of:
individually accessing each of said unit cells; and
25 controlling charge-integration of each said unit cell, independently of said charge-integration of other said plurality of unit cells.
20. A method according to claim 19, further comprising the steps of :
determining the charge-integration time for each said unit cell; and
programming each said unit cell according to said determined
30 charge-integration time.

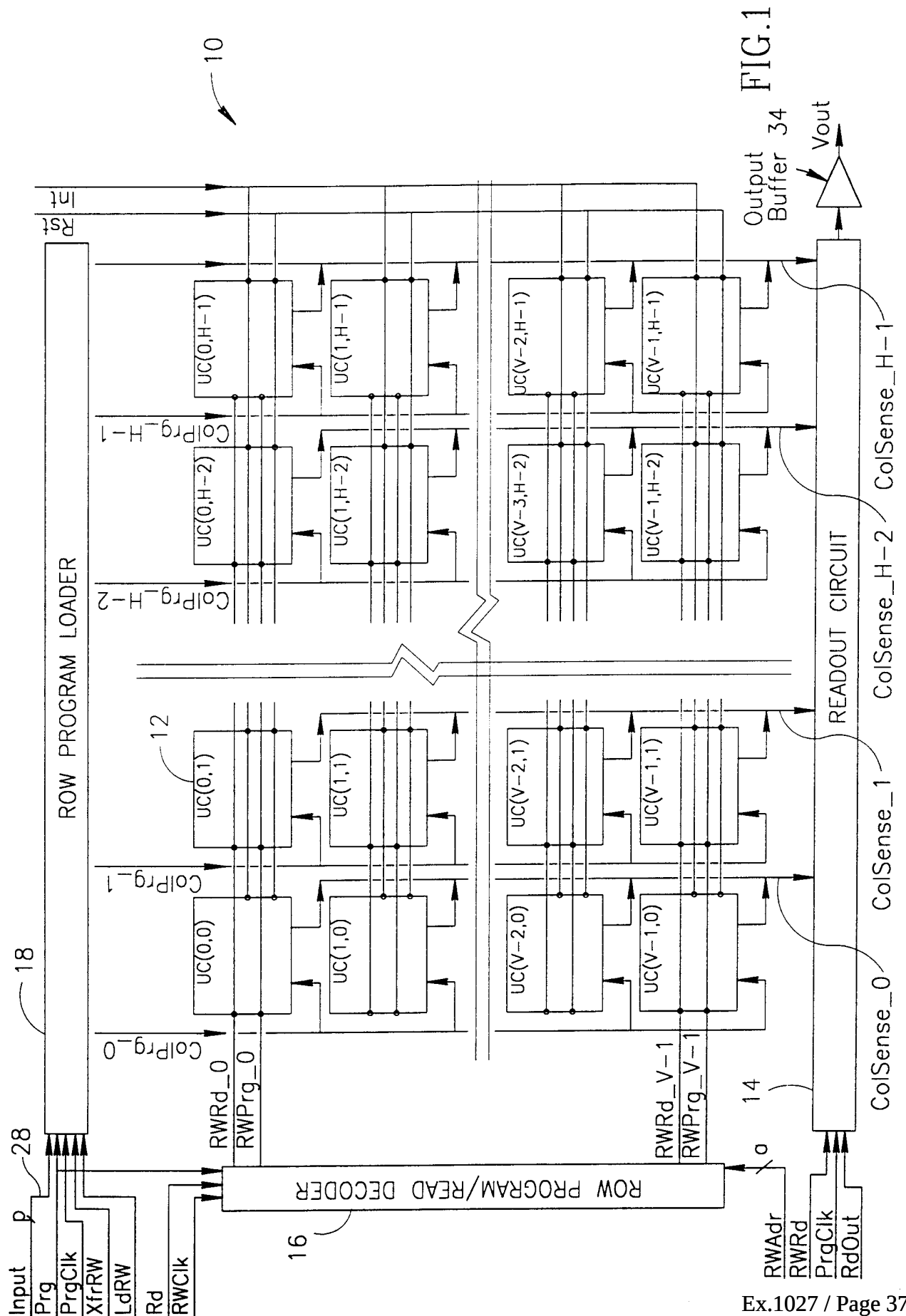
21. A method for improving the intra-scene dynamic range of an image-sensor array of multiple unit cells, the method comprising the steps of:
individually accessing each said unit cell; and
individually controlling each said unit cell.
- 5 22. A method according to claim 21, wherein said step of individually controlling comprises the step of:
individually controlling the charge-integration time of each said unit cells.
23. A method according to claim 21, wherein said step of individually controlling said charge-integration time comprises the step of:
10 individually programming each said unit cell.
24. A method according to claim 23, wherein said step of individually programming each said unit cell comprises the step of:
programming each said unit cell according to a pre-determined charge-integration time.
- 15 25. A method according to claim 23, wherein said step of individually programming each said unit cell comprises the step of:
programming each said unit cell with multiple charge-integration sub-periods.
26. A method for improving the intra-scene dynamic range of an image-sensor array of multiple unit cells, said method comprising the steps of :
20 individually accessing each said unit cell;
individually programming each said unit cell; and
individually controlling the charge-integration of each said unit cell,
wherein steps of accessing, programming and controlling are performed at video rate.
- 25 27. A programmable image sensor comprising a multiplicity of unit cells for capturing an image, the sensor comprising:
a first plurality P of input lines for carrying data;
a second plurality H of columns which are connected to said cells , wherein P is equal or smaller than H; and
30 a controller for receiving said data and selectively distributing said data to said columns, to program N times said array, within a single frame of motion video.

28. A programmable image sensor according to claim 27, wherein said data is programming data.

29. A programmable image sensor according to claim 28, wherein said programming data comprises charge-integration/non-integration state data for
5 each of said plurality of unit cells.

30. A programmable image sensor according to claim 28, wherein each of said plurality of programmable unit cells is individually controlled.

1/14



2/14

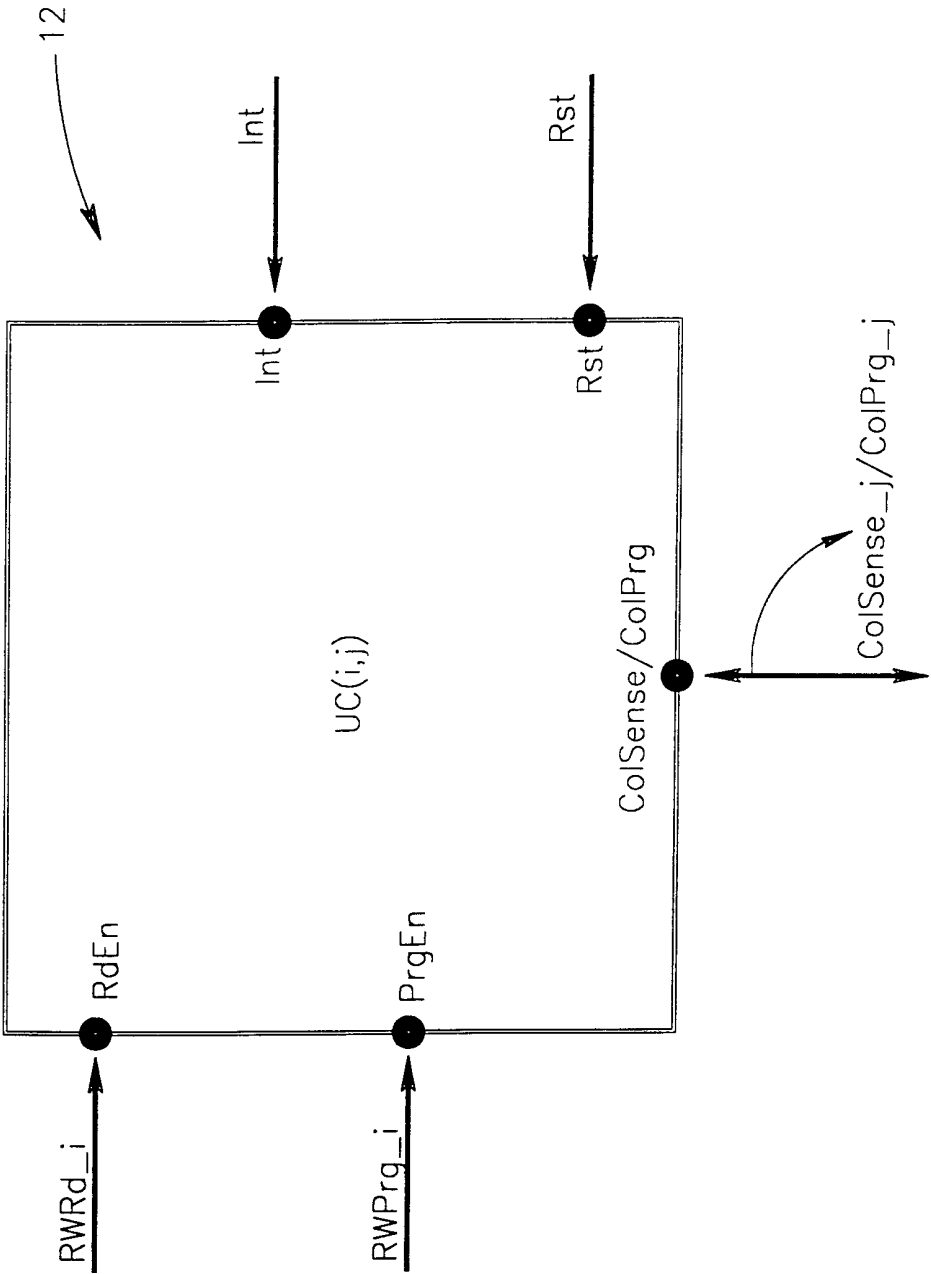


FIG.2

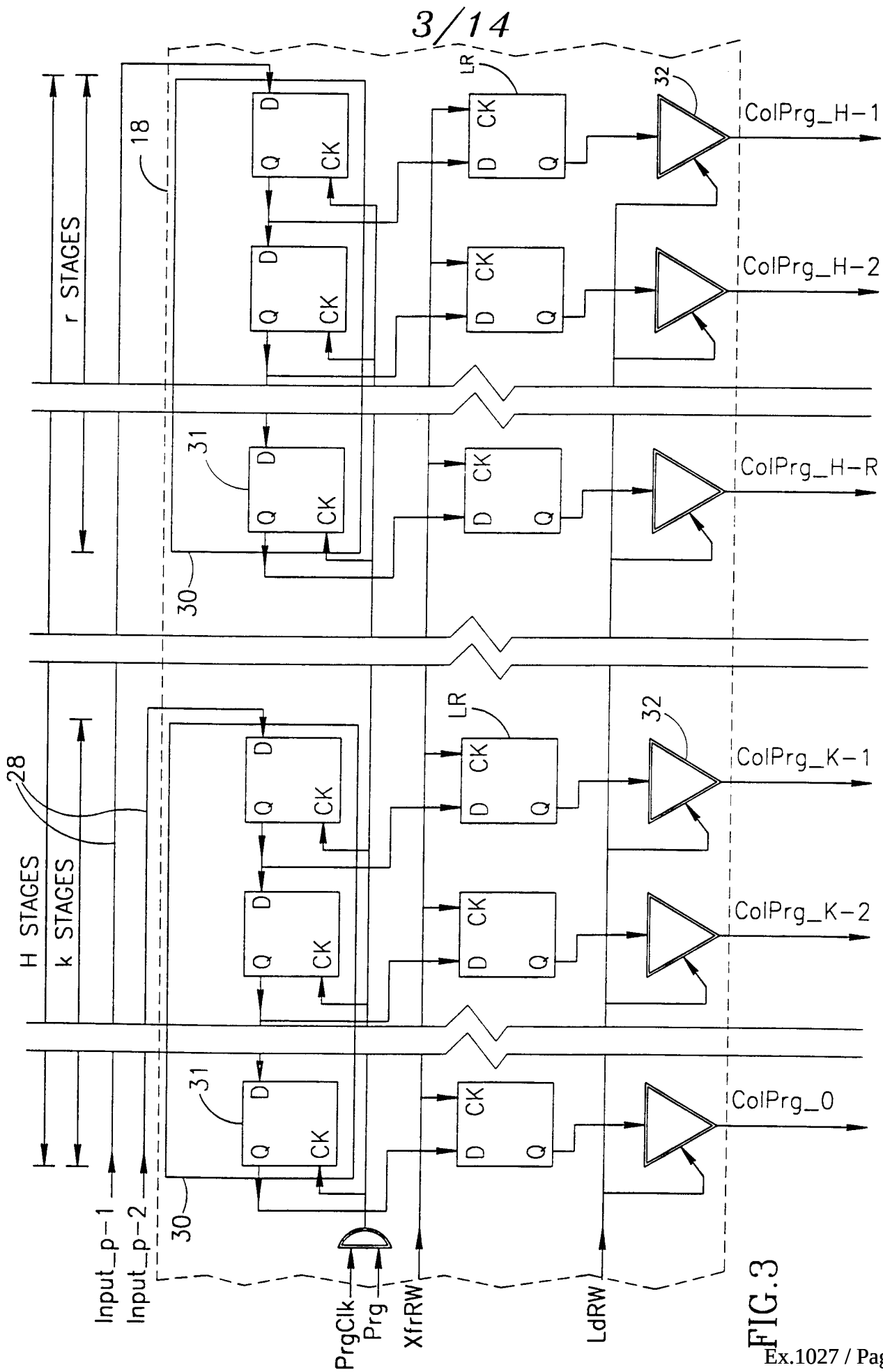


FIG. 3

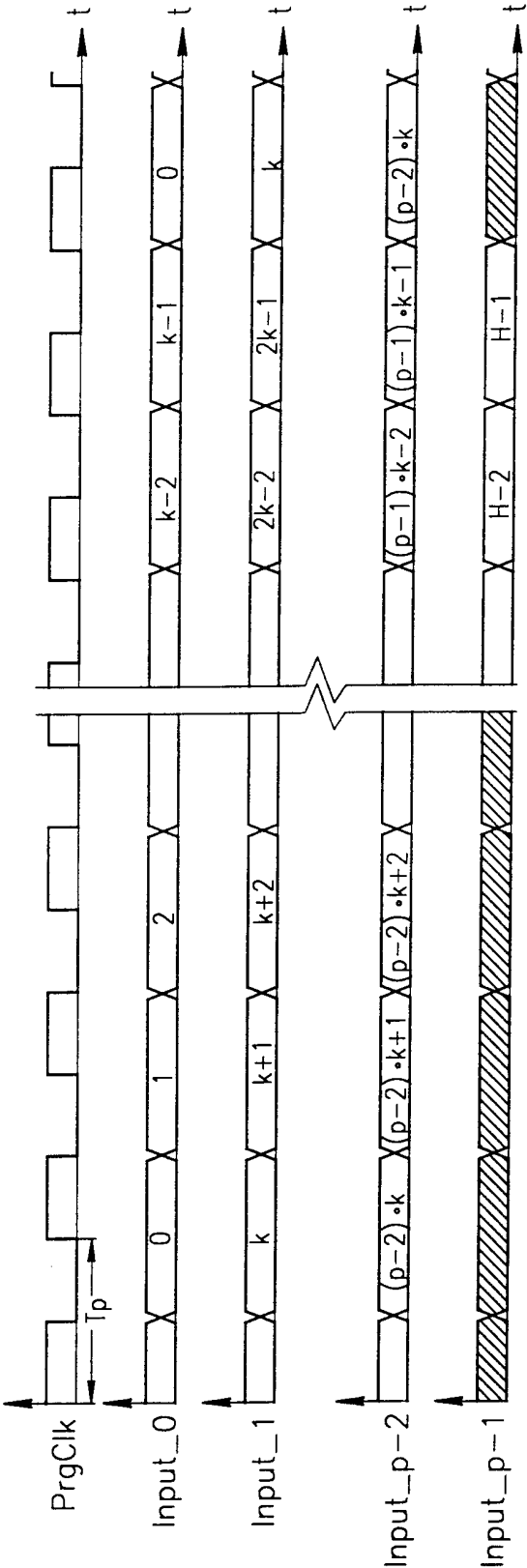


FIG.4A

5/14

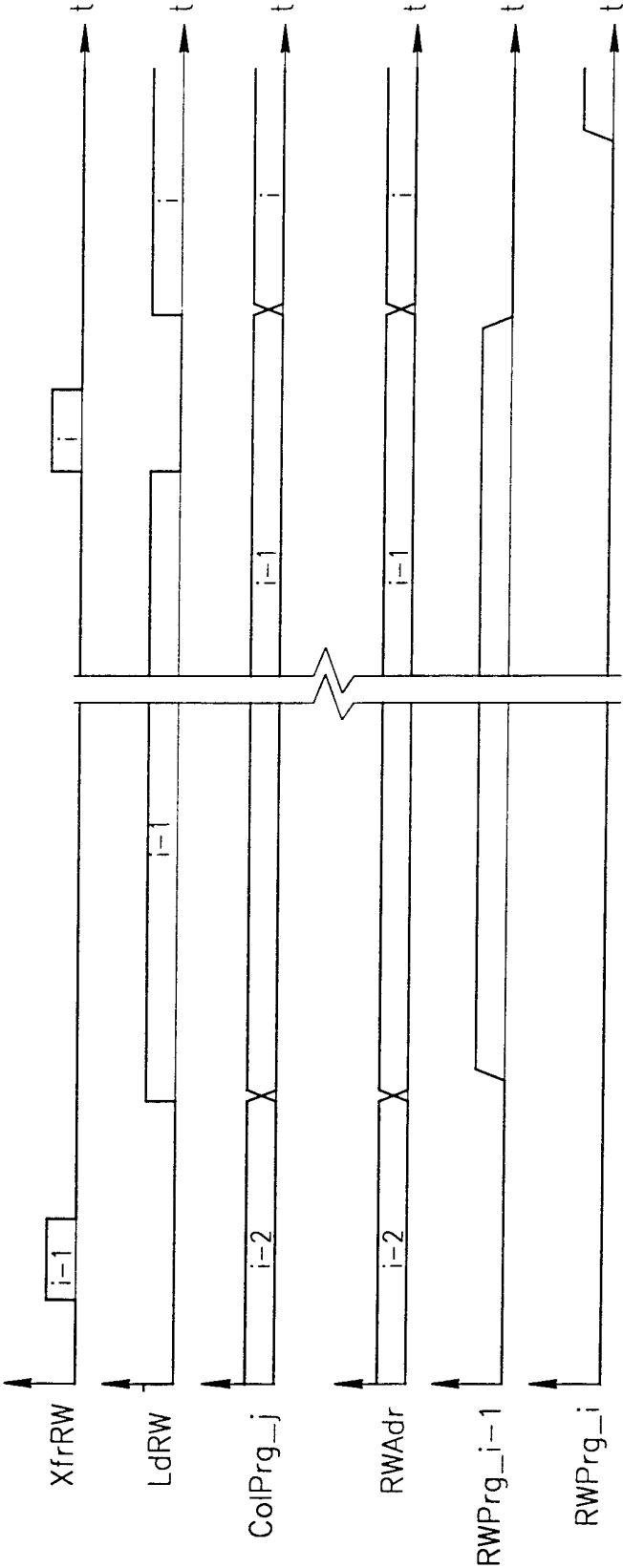
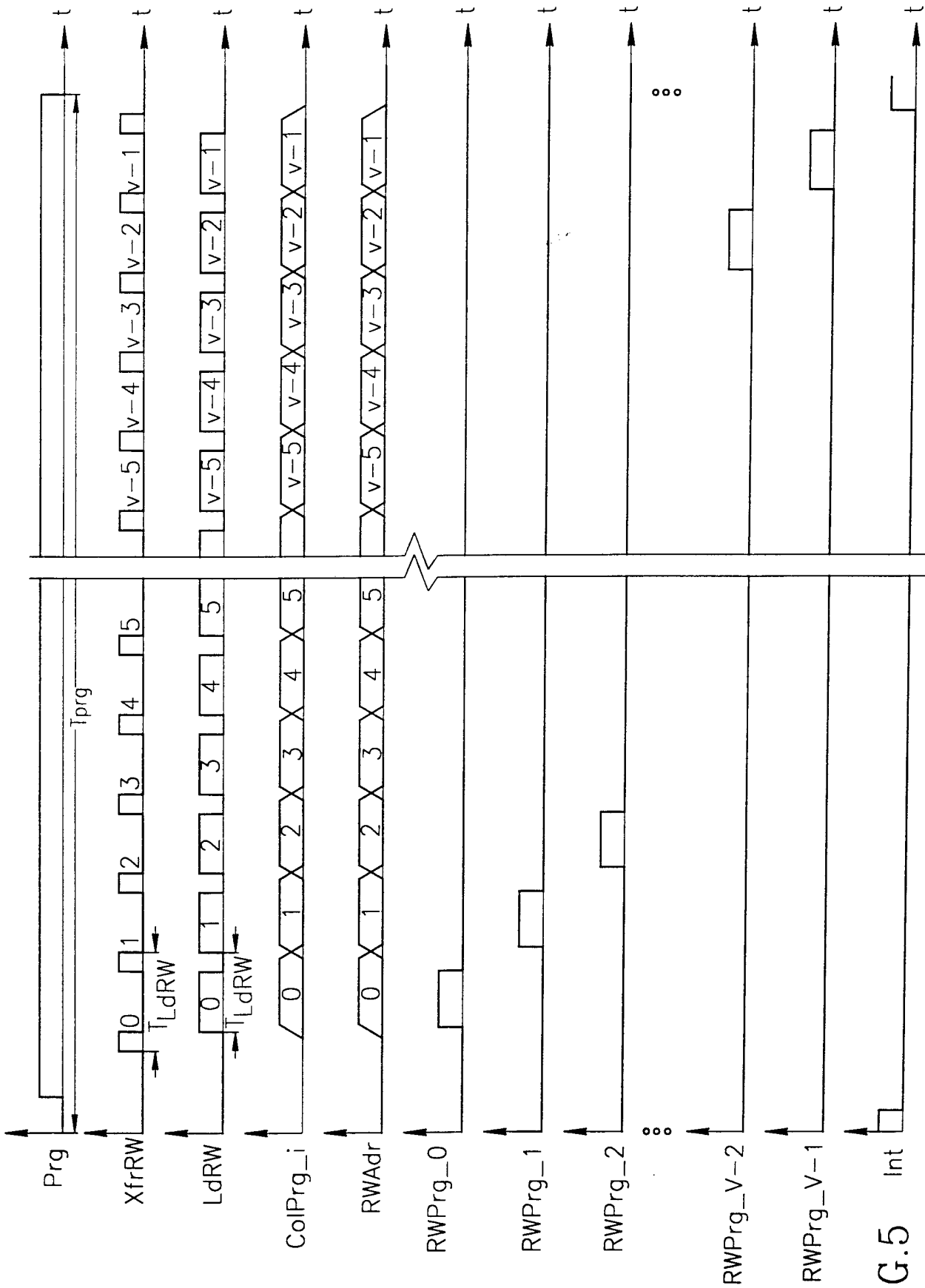


FIG.4B

6/14



7/14

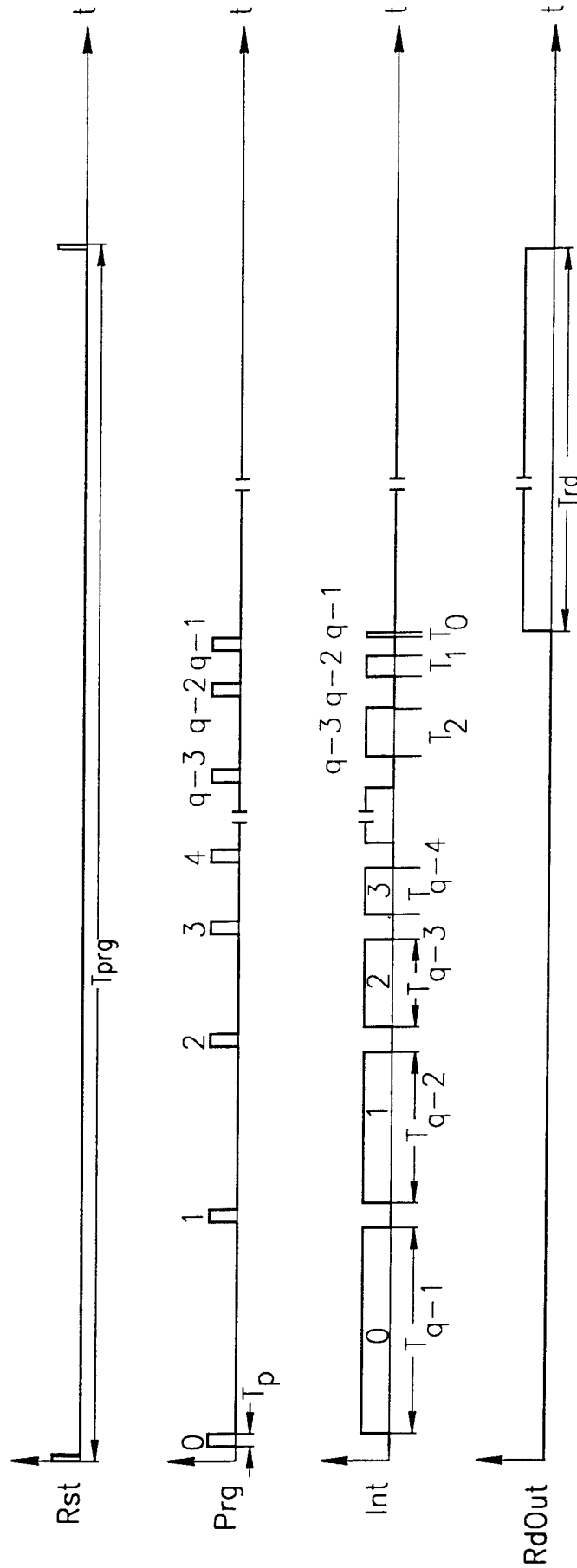


FIG.6

8/14

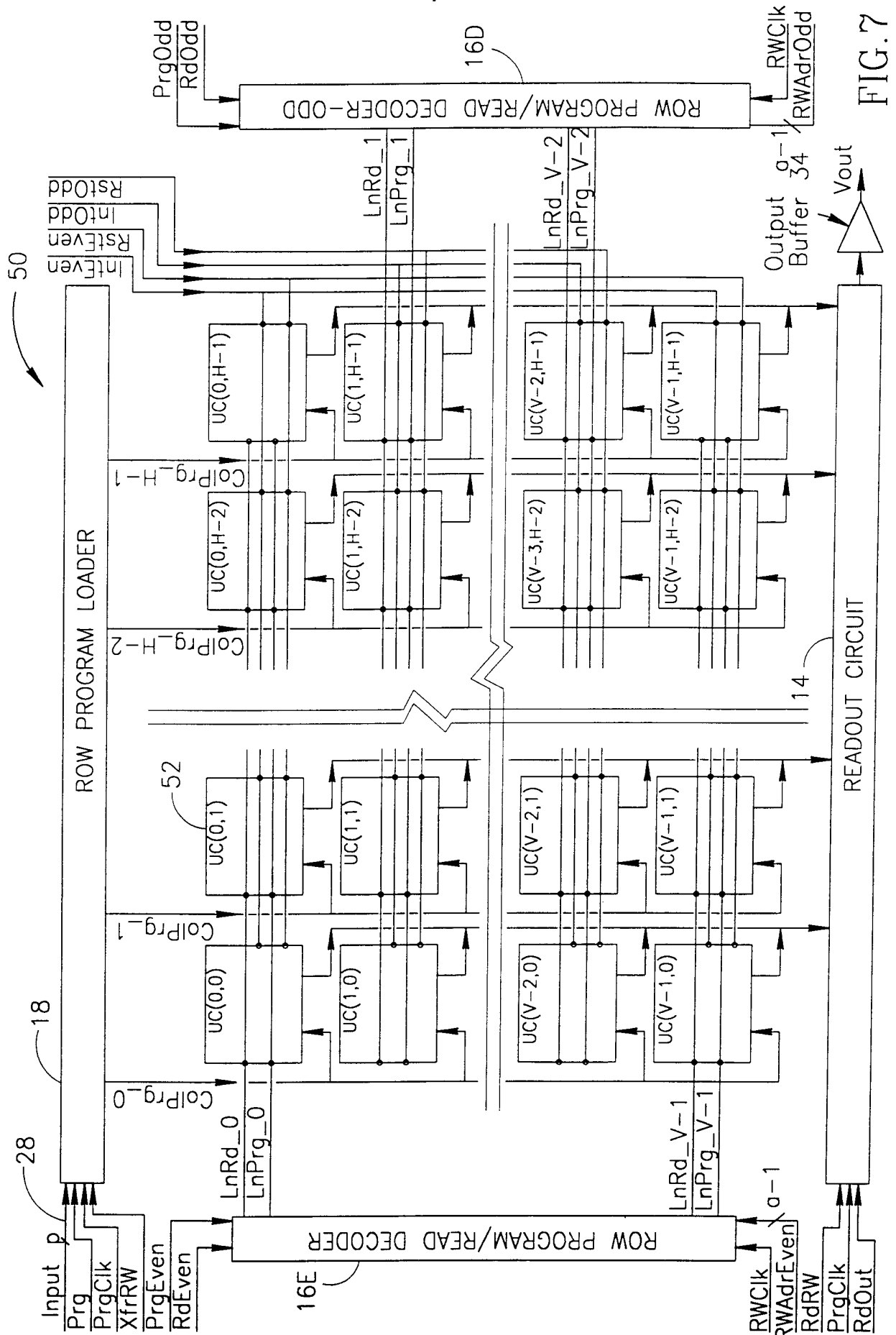


FIG. 7

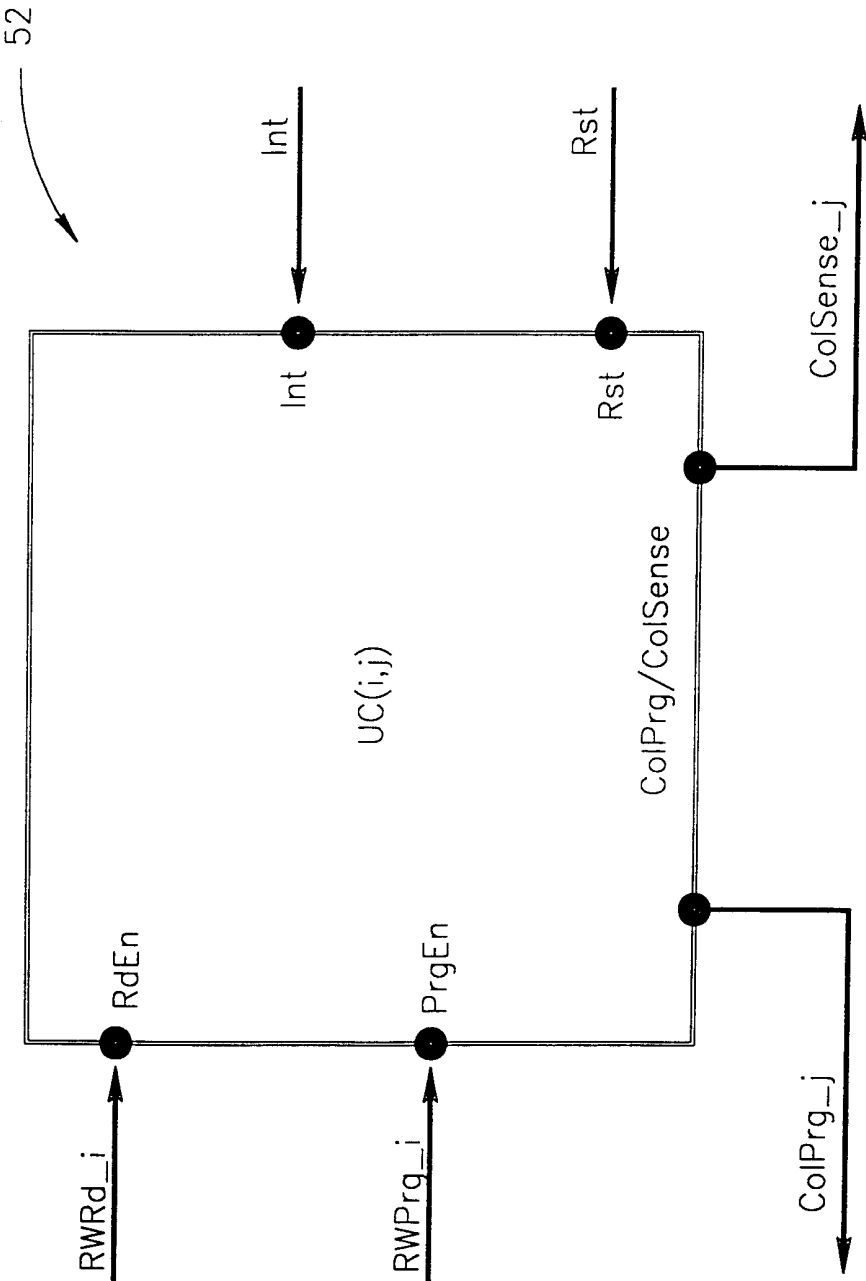


FIG.8

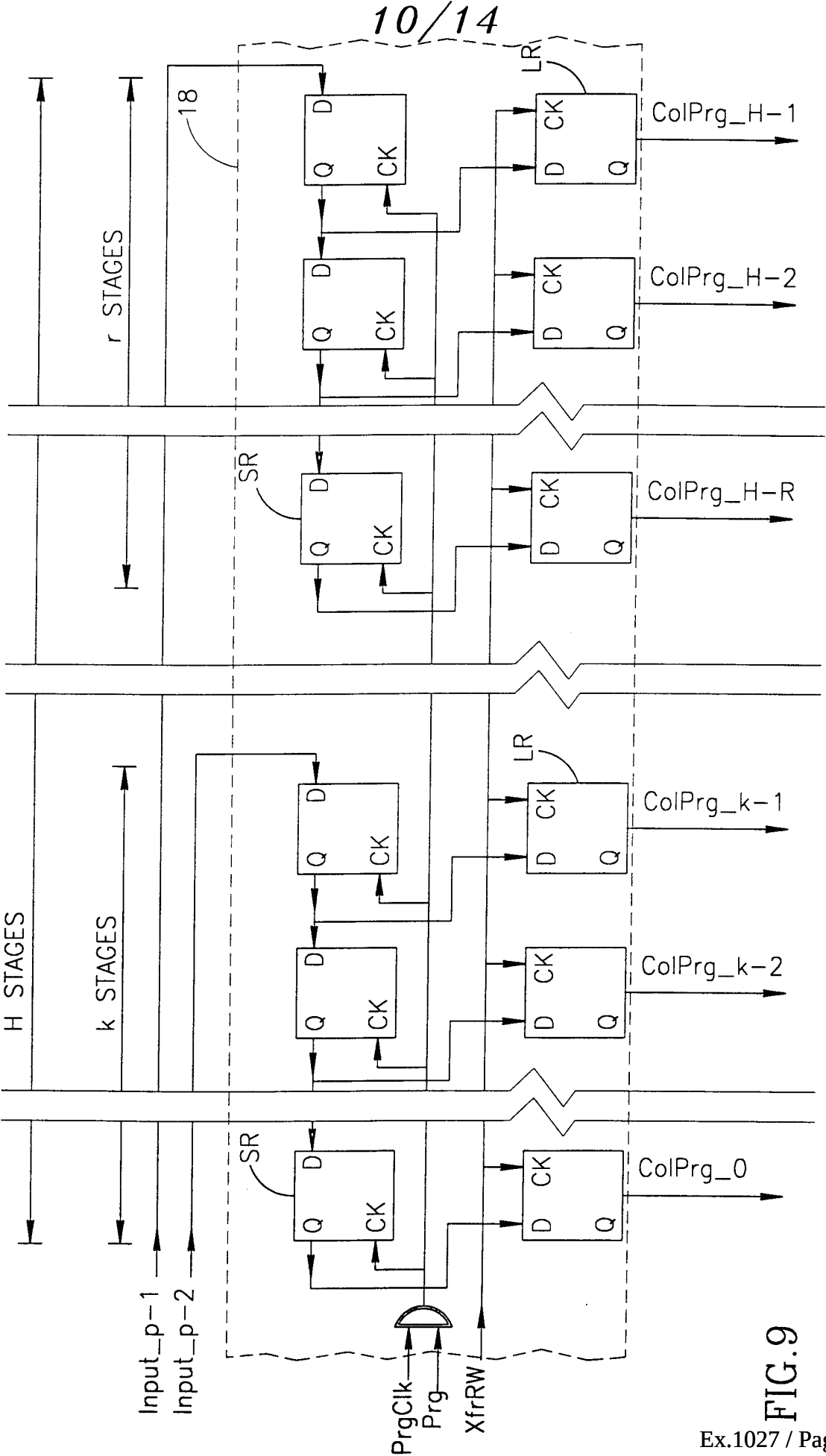


FIG.9

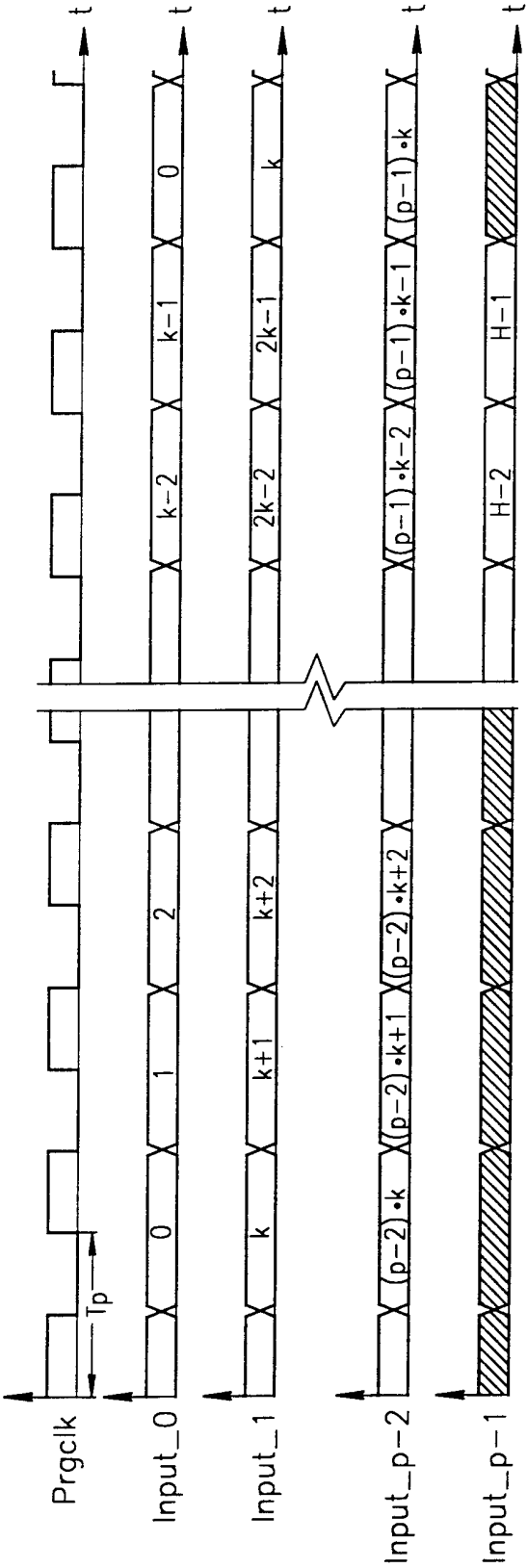


FIG.10A

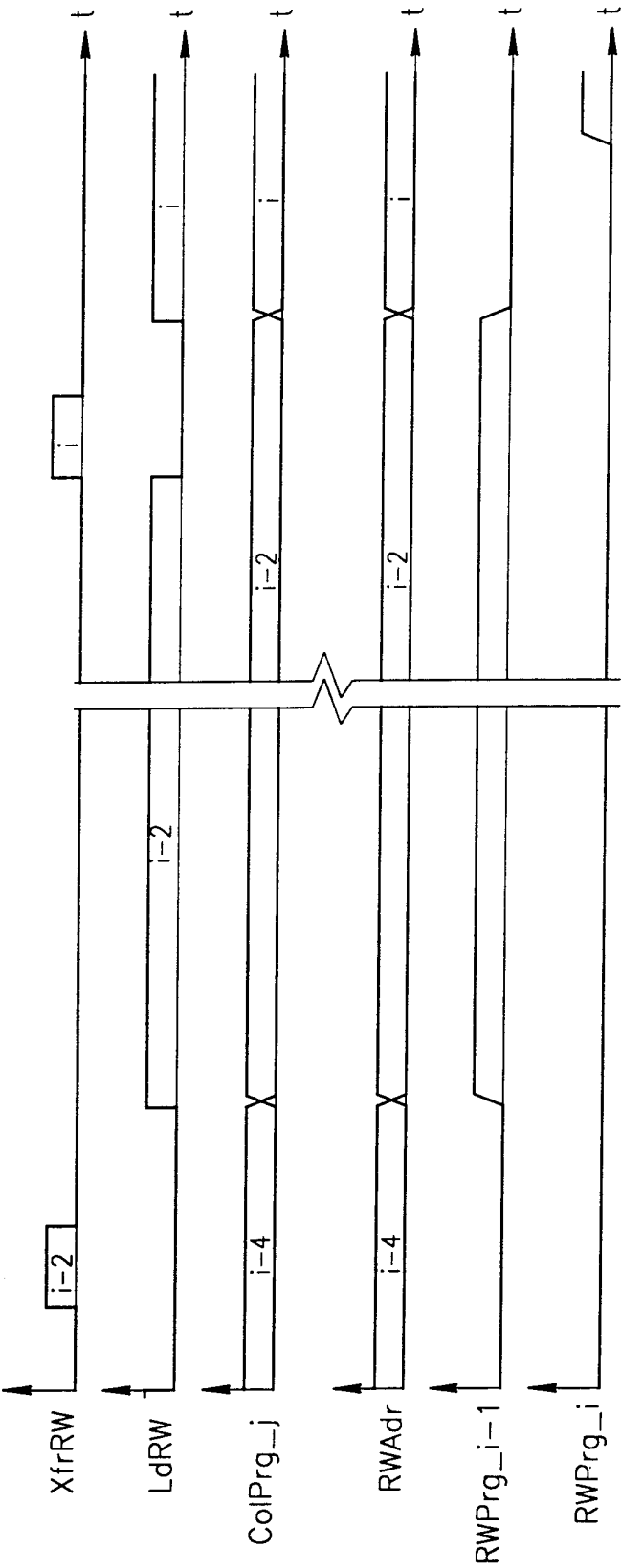


FIG.10B

13/14

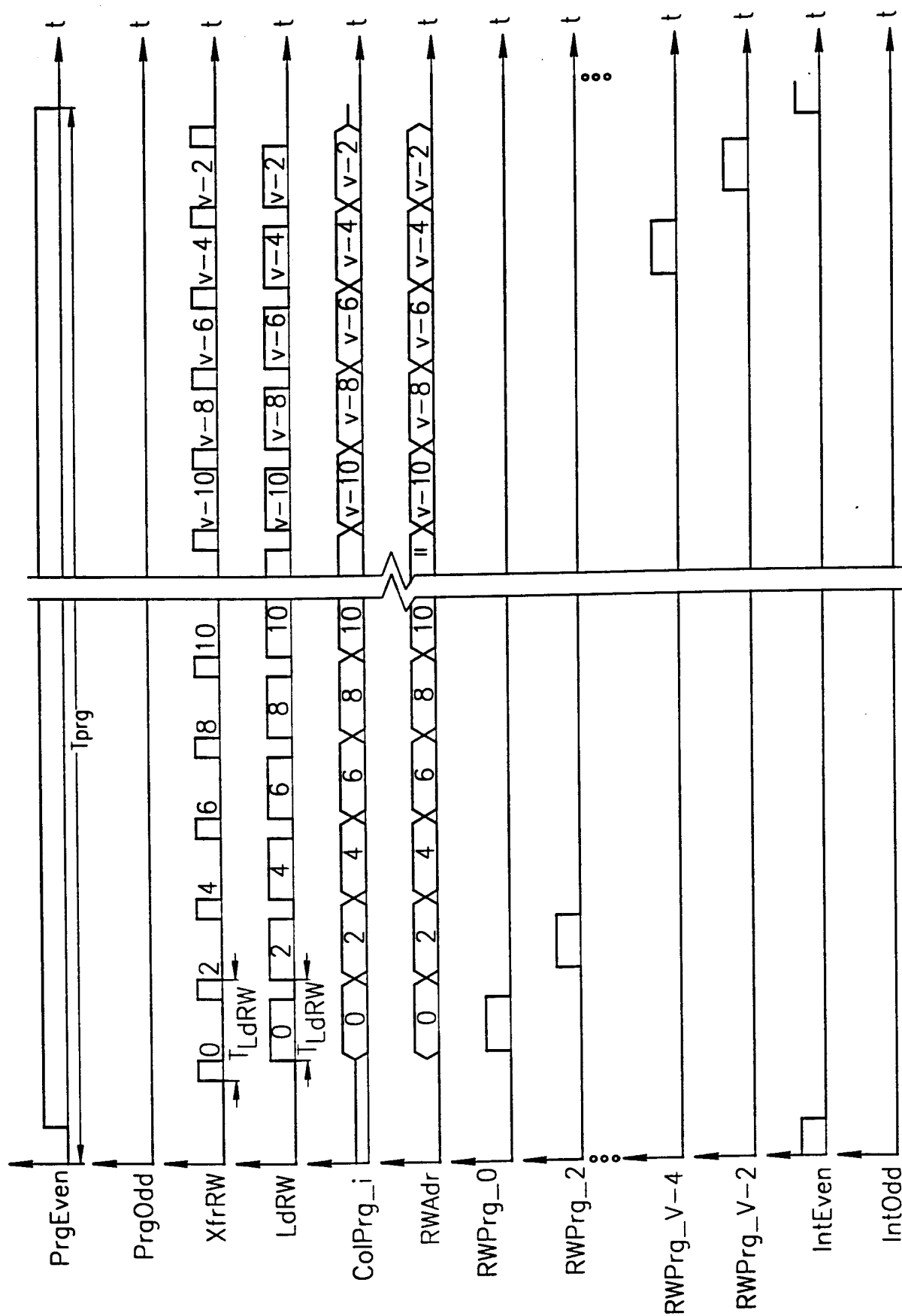


FIG.11

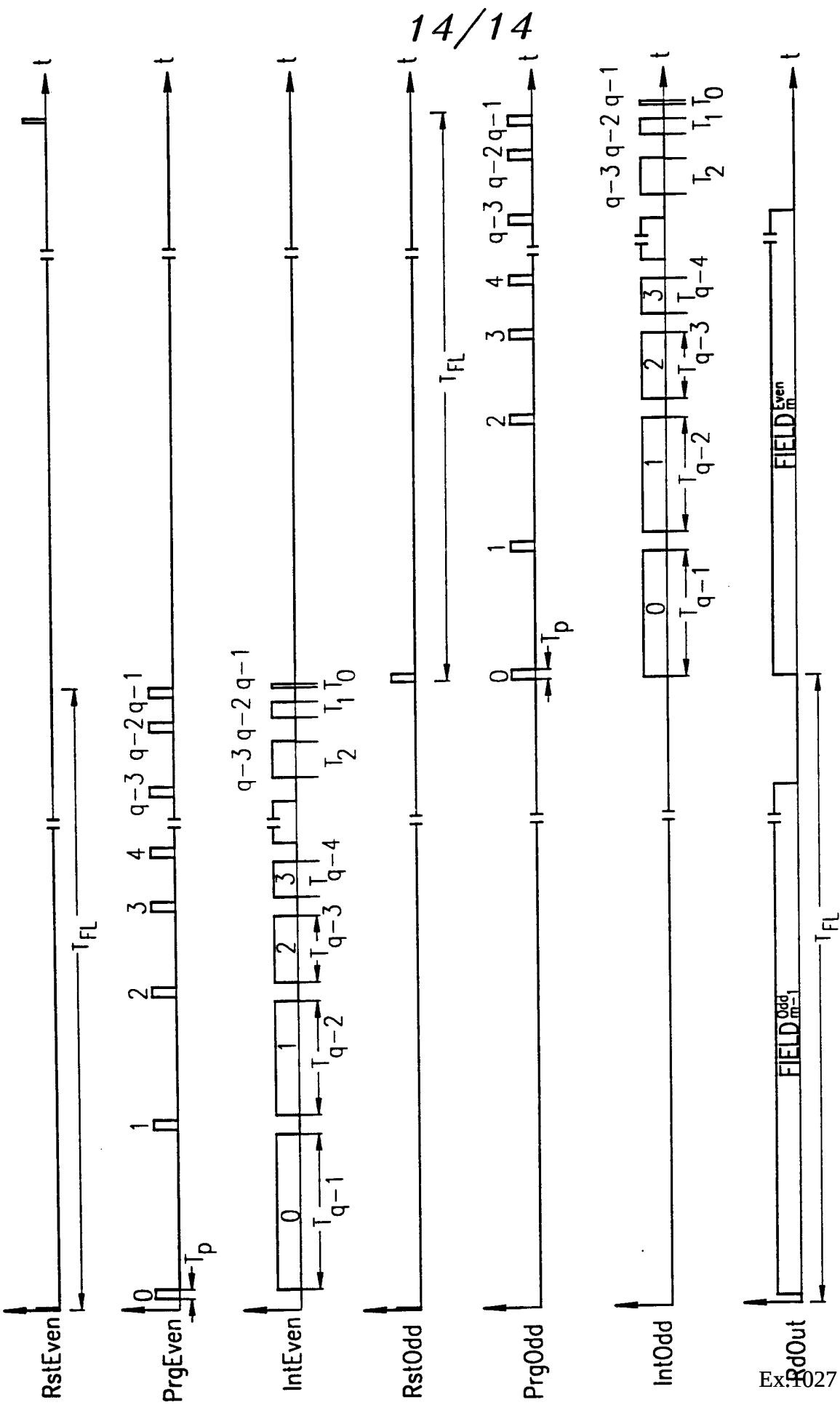


FIG.12