(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 2 301 753 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
30.03.2011 Bulletin 2011/13

(21) Application number: **10193974.2**

(22) Date of filing: **27.05.2004**

(51) Int Cl.:
B41J 2/07 *(2006.01)* ***B41J 2/155*** *(2006.01)*
B41J 2/515 *(2006.01)*

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PL PT RO SE SI SK TR

(62) Document number(s) of the earlier application(s) in accordance with Art. 76 EPC:
04734974.1 / 1 765 595

(71) Applicant: **Silverbrook Research Pty. Ltd**
Balmain, New South Wales 2041 (AU)

(72) Inventors:
- **Walmsley, Simon R. W.**
  Balmain, New South Wales 2041 (AU)
- **Silverbrook, Kia**
  Balmain, New South Wales 2041 (AU)
- **Jackson Pulver, Mark**
  Balmain, New South Wales 2041 (AU)
- **Sheahan, John R.**
  Balmain, New South Wales 2041 (AU)
- **Plunkett, Richard T.**
  Balmain, New South Wales 2041 (AU)
- **Webb, Michael J.**
  Balmain, New South Wales 2041 (AU)
- **Morphett, Benjamin D.**
  Balmain, New South Wales 2041 (AU)

(74) Representative: **Keane, Paul Fachtna et al**
Hanna, Moore & Curley
13 Lower Lad Lane
2 Dublin (IE)

Remarks:
This application was filed on 07-12-2010 as a divisional application to the application mentioned under INID code 62.

(54) Printhead module having a dropped row and printer controller for supplying data thereto

(57) A printhead module includes at least one row of printhead nozzles. Each row includes at least one displaced row portion. The displacement of the row portion includes a component in a direction normal to that of a pagewidth to be printed.

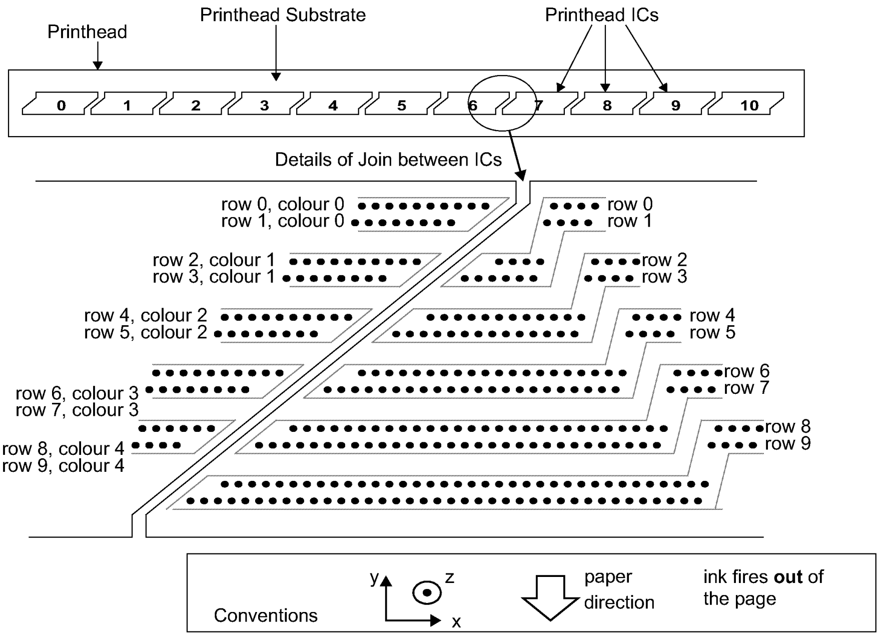

FIG. 8

EP 2 301 753 A1

Printed by Jouve, 75001 PARIS (FR)

Description

FIELD OF THE INVENTION

[0001] The present invention relates to pagewidth inkjet printers.

BACKGROUND

[0002] Pagewidth printheads may be constructed from multiple printhead chips butted together. Accordingly, multiple identical printhead chips must be capable of being linked together to form an effectively horizontal assembled printhead.

SUMMARY OF THE INVENTION

[0003] The present invention provides a printhead module, a printhead and a printer controller as defined hereinbelow in the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004]

Figure 1. Single SoPEC A4 Simplex system

Figure 2. Dual SoPEC A4 Simplex system

Figure 3. Dual SoPEC A4 Duplex system

Figure 4. Dual SoPEC A3 simplex system

Figure 5. Quad SoPEC A3 duplex system

Figure 6. SoPEC A4 Simplex system with extra SoPEC used as DRAM storage

Figure 7. SoPEC A4 Simplex system with network connection to Host PC

Figure 8. Print construction and Nozzle position

Figure 9. Conceptual horizontal misplacement between segments

Figure 10. Printhead row positioning and default row firing order

Figure 11. Firing order of fractionally misaligned segment

Figure 12. Example of yaw in printhead IC misplacement

Figure 13. Vertical nozzle spacing

Figure 14. Single printhead chip plus connection to second chip

Figure 15. Two printheads connected to form a larger printhead

Figure 16. Colour arrangement

Figure 17. Nozzle Offset at Linking Ends

Figure 18. Bonding Diagram

Figure 19. Block Diagram

Figure 20. TDC block diagram

Figure 21. TDC waveform

Figure 22. TDC construction

Detailed Description of Preferred Embodiment

[0005] Various aspects of the preferred and other embodiments will now be described.

[0006] It will be appreciated that the following description is a highly detailed exposition of the hardware and associated methods that together provide a printing system capable of relatively high resolution, high speed and low cost printing compared to prior art systems.

[0007] Much of this description is based on technical design documents, so the use of words like "must", "should" and "will", and all others that suggest limitations or positive attributes of the performance of a particular product, should not be interpreted as applying to the invention in general. These comments, unless clearly referring to the invention in general, should be considered as desirable or intended features in a particular design rather than a requirement of the invention. The intended scope of the invention is defined in the claims.

[0008] Also throughout this description, "printhead module" and "printhead" are used somewhat interchangeably. Technically, a "printhead" comprises one or more "printhead modules", but occasionally the former is used to refer to the latter. It should be clear from the context which meaning should be allocated to any use of the word "printhead".

[0009] In general:

Linking Printhead Refers to a page-width printhead constructed from multiple linking printhead ICs

Linking Printhead IC A MEMS IC. Multiple ICs link together to form a complete printhead. An A4/Letter page width printhead requires 11 printhead ICs.

1 Introduction

[0010] The SoPEC ASIC (Small office home office Print Engine Controller) is suitable for use in price sensitive SoHo printer products. The SoPEC ASIC is intended to be a relatively low cost solution for linking printhead control, replacing the multichip solutions in larger more professional systems with a single chip. The increased cost competitiveness is achieved by integrating several systems such as a modified PEC1 printing pipeline, CPU control system, peripherals and memory sub-system onto one SoC ASIC, reducing component count and simplifying board design. SoPEC contains features making it suitable for multifunction or "all-in-one" devices as well as dedicated printing systems.

[0011] This section will give a general introduction to Memjet printing systems, introduce the components that make a linking printhead system, describe a number of system architectures and show how several SoPECs can be used to achieve faster, wider and/or duplex printing. The section "SoPEC ASIC" describes the SoC SoPEC ASIC, with subsections describing the CPU, DRAM and Print Engine Pipeline subsystems. Each section gives a detailed description of the blocks used and their operation within the overall print system.

[0012] Basic features of the preferred embodiment of SoPEC include:

- Continuous 30ppm operation for 1600dpi output at A4/Letter.
- Linearly scalable (multiple SoPECs) for increased print speed and/or page width.
- 192MHz internal system clock derived from low-speed crystal input
- PEP processing pipeline, supports up to 6 color channels at 1 dot per channel per clock cycle
- Hardware color plane decompression, tag rendering, halftoning and compositing
- Data formatting for Linking Printhead
- Flexible compensation for dead nozzles, printhead misalignment etc.
- Integrated 20Mbit (2.5MByte) DRAM for print data and CPU program store
- LEON SPARC v8 32-bit RISC CPU
- Supervisor and user modes to support multi-threaded software and security
- 1kB each of I-cache and D-cache, both direct mapped, with optimized 256-bit fast cache update.
- 1 x USB2.0 device port and 3 x USB2.0 host ports (including integrated PHYs)
- Support high speed (480Mbit/sec) and full speed (12Mbit/sec) modes of USB2.0
- Provide interface to host PC, other SoPECs, and external devices e.g. digital camera
- Enable alternative host PC interfaces e.g. via external USB/ethernet bridge
- Glueless high-speed serial LVDS interface to multiple Linking Printhead chips

- 64 remappable GPIOs, selectable between combinations of integrated system control components:
- 2 x LSS interfaces for QA chip or serial EEPROM
- LED drivers, sensor inputs, switch control outputs
- Motor controllers for stepper and brushless DC motors
- Microprogrammed multi-protocol media interface for scanner, external RAM/Flash, etc.
- 112-bit unique ID plus 112-bit random number on each device, combined for security protocol support
- IBM Cu-11 0.13 micron CMOS process, 1.5V core supply, 3.3V 10.
- 208 pin Plastic Quad Flat Pack

2 Print Quality Considerations

[0013] The preferred embodiment linking printhead produces 1600 dpi bi-level dots. On low-diffusion paper, each ejected drop forms a 22.5 m diameter dot. Dots are easily produced in isolation, allowing dispersed-dot dithering to be exploited to its fullest. Since the preferred form of the linking printhead is pagewidth and operates with a constant paper velocity, color planes are printed in good registration, allowing dot-on-dot printing. Dot-on-dot printing minimizes 'muddying' of midtones caused by inter-color bleed.

[0014] A page layout may contain a mixture of images, graphics and text. Continuous-tone (contone) images and graphics are reproduced using a stochastic dispersed-dot dither. Unlike a clustered-dot (or amplitude-modulated) dither, a *dispersed-dot* (or frequency-modulated) dither reproduces high spatial frequencies (i.e. image detail) almost to the limits of the dot resolution, while simultaneously reproducing lower spatial frequencies to their full color depth, when spatially integrated by the eye. A *stochastic* dither matrix is carefully designed to be free of objectionable low-frequency patterns when tiled across the image. As such its size typically exceeds the minimum size required to support a particular number of intensity levels (e.g. 16 16 8 bits for 257 intensity levels).

[0015] Human contrast sensitivity peaks at a spatial frequency of about 3 cycles per degree of visual field and then falls off logarithmically, decreasing by a factor of 100 beyond about 40 cycles per degree and becoming immeasurable beyond 60 cycles per degree. At a normal viewing distance of 12 inches (about 300mm), this translates roughly to 200-300 cycles per inch (cpi) on the printed page, or 400-600 samples per inch according to Nyquist's theorem.

[0016] In practice, contone resolution above about 300 ppi is of limited utility outside special applications such as medical imaging. Offset printing of magazines, for example, uses contone resolutions in the range 150 to 300 ppi. Higher resolutions contribute slightly to color error through the dither.

[0017] Black text and graphics are reproduced directly using bi-level black dots, and are therefore not anti-aliased (i.e. low-pass filtered) before being printed. Text should therefore be *supersampled* beyond the perceptual limits discussed above, to produce smoother edges when spatially integrated by the eye. Text resolution up to about 1200 dpi continues to contribute to perceived text sharpness (assuming low-diffusion paper).

[0018] A Netpage printer, for example, may use a contone resolution of 267 ppi (i.e. 1600 dpi 6), and a black text and graphics resolution of 800 dpi. A high end office or departmental printer may use a contone resolution of 320 ppi (1600 dpi / 5) and a black text and graphics resolution of 1600 dpi. Both formats are capable of exceeding the quality of commercial (offset) printing and photographic reproduction.

3 Memjet Printer Architecture

[0019] The SoPEC device can be used in several printer configurations and architectures.

[0020] In the general sense, every preferred embodiment SoPEC-based printer architecture will contain:

- One or more SoPEC devices.
- One or more linking printheads.
- Two or more LSS busses.
- Two or more QA chips.
- Connection to host, directly via USB2.0 or indirectly.
- Connections between SoPECs (when multiple SoPECs are used).

[0021] Some example printer configurations as outlined in Section 4.2. The various system components are outlined briefly in Section 4.1.

4.1 System Components

4.1.1 SoPEC Print Engine Controller

[0022] The SoPEC device contains several system on a chip (SoC) components, as well as the print engine pipeline control application specific logic.

4.1.1.1 Print Engine Pipeline (PEP) Logic

[0023] The PEP reads compressed page store data from the embedded memory, optionally decompresses the data and formats it for sending to the printhead. The print engine pipeline functionality includes expanding the page image, dithering the contone layer, compositing the black layer over the contone layer, rendering of Netpage tags, compensation for dead nozzles in the printhead, and sending the resultant image to the linking printhead.

4.1.1.2 Embedded CPU

[0024] SoPEC contains an embedded CPU for general-purpose system configuration and management. The CPU performs page and band header processing, motor control and sensor monitoring (via the GPIO) and other system control functions. The CPU can perform buffer management or report buffer status to the host. The CPU can optionally run vendor application specific code for general print control such as paper ready monitoring and LED status update.

4.1.1.3 Embedded Memory Buffer

[0025] A 2.5Mbyte embedded memory buffer is integrated onto the SoPEC device, of which approximately 2Mbytes are available for compressed page store data. A compressed page is divided into one or more bands, with a number of bands stored in memory. As a band of the page is consumed by the PEP for printing a new band can be downloaded. The new band may be for the current page or the next page.

[0026] Using banding it is possible to begin printing a page before the complete compressed page is downloaded, but care must be taken to ensure that data is always available for printing or a buffer underrun may occur.

[0027] A Storage SoPEC acting as a memory buffer (Section 4.2.6) could be used to provide guaranteed data delivery.

4.1.1.4 Embedded USB2.0 Device Controller

[0028] The embedded single-port USB2.0 device controller can be used either for interface to the host PC, or for communication with another SoPEC as an ISCSlave. It accepts compressed page data and control commands from the host PC or ISCMaster SoPEC, and transfers the data to the embedded memory for printing or downstream distribution.

4.1.1.5 Embedded USB2.0 Host Controller

[0029] The embedded three-port USB2.0 host controller enables communication with other SoPEC devices as a ISCMaster, as well as interfacing with external chips (e.g. for Ethernet connection) and external USB devices, such as digital cameras.

4.1.1.6 Embedded Device/Motor Controllers

[0030] SoPEC contains embedded controllers for a variety of printer system components such as motors, LEDs etc, which are controlled via SoPEC's GPIOs. This minimizes the need for circuits external to SoPEC to build a complete printer system.

4.1.2 Linking Printhead

[0031] The printhead is constructed by abutting a number of printhead ICs together. Each SoPEC can drive up to 12 printhead ICs at data rates up to 30ppm or 6 printhead ICs at data rates up to 60ppm. For higher data rates, or wider printheads, multiple SoPECs must be used.

4.1.3 LSS interface bus

[0032] Each SoPEC device has 2 LSS system buses for communication with QA devices for system authentication and ink usage accounting. The number of QA devices per bus and their position in the system is unrestricted with the exception that *PRINTER_QA* and *INK_QA* devices should be on separate LSS busses.

4.1.4 QA devices

[0033] Each SoPEC system can have several QA devices. Normally each printing SoPEC will have an associated *PRINTER_QA.* Ink cartridges will contain an *INK_QA* chip. *PRINTER_QA* and *INK_QA* devices should be on separate LSS busses. All QA chips in the system are physically identical with flash memory contents defining *PRINTER_QA* from *INK _QA* chip.

4.1.5 Connections between SoPECs

[0034] In a multi-SoPEC system, the primary communication channel is from a USB2.0 Host port on one SoPEC (the ISCMaster), to the USB2.0 Device port of each of the other SoPECs (ISCSlaves). If there are more ISCSlave SoPECs than available USB Host ports on the ISCMaster, additional connections could be via a USB Hub chip, or daisy-chained SoPEC chips. Typically one or more of SoPEC's GPIO signals would also be used to communicate specific events between multiple SoPECs.

4.1.6 Non-USB Host PC Communication

[0035] The communication between the host PC and the ISCMaster SoPEC may involve an external chip or subsystem, to provide a non-USB host interface, such as ethernet or WiFi. This subsystem may also contain memory to provide an additional buffered band/page store, which could provide guaranteed bandwidth data deliver to SoPEC during complex page prints.

4.2 *Possible SoPEC Systems*

[0036] Several possible SoPEC based system architectures exist. The following sections outline some possible architectures. It is possible to have extra SoPEC devices in the system used for DRAM storage. The QA chip configurations shown are indicative of the flexibility of LSS bus architecture, but not limited to those configurations.

4.2.1 A4 Simplex at 30 ppm with 1 SoPEC device

[0037] In Figure 1, a single SoPEC device is used to control a linking printhead with 11 printhead ICs. The SoPEC receives compressed data from the host through its USB device port. The compressed data is processed and transferred to the printhead. This arrangement is limited to a speed of 30ppm. The single SoPEC also controls all printer components such as motors, LEDs, buttons etc, either directly or indirectly.

4.2.2 A4 Simplex at 60 ppm with 2 SoPEC devices

[0038] In Figure 2, two SoPECs control a single linking printhead, to provide 60ppm A4 printing. Each SoPEC drives 5 or 6 of the printheads ICs that make up the complete printhead. SoPEC #0 is the ISCMaster, SoPEC #1 is an ISCSlave. The ISCMaster receives all the compressed page data for both SoPECs and re-distributes the compressed data for the ISCSlave over a local USB bus. There is a total of 4MBytes of page store memory available if required. Note that, if each page has 2MBytes of compressed data, the USB2.0 interface to the host needs to run in high speed (not full speed) mode to sustain 60ppm printing. (In practice, many compressed pages will be much smaller than 2MBytes). The control of printer components such as motors, LEDs, buttons etc, is shared between the 2 SoPECs in this configuration.

4.2.3 A4 Duplex with 2 SoPEC devices

[0039] In Figure 3, two SoPEC devices are used to control two printheads. Each printhead prints to opposite sides of the same page to achieve duplex printing. SoPEC #0 is the ISCMaster, SoPEC #1 is an ISCSlave. The ISCMaster receives all the compressed page data for both SoPECs and re-distributes the compressed data for the ISCSlave over a local USB bus. This configuration could print 30 double-sided pages per minute.

4.2.4 A3 Simplex with 2 SoPEC devices

[0040] In Figure 4, two SoPEC devices are used to control one A3 linking printhead, constructed from 16 printhead

ICs. Each SoPEC controls 8 printhead ICs. This system operates in a similar manner to the 60ppm A4 system in Figure 2, although the speed is limited to 30ppm at A3, since each SoPEC can only drive 6 printhead ICs at 60ppm speeds. A total of 4Mbyte of page store is available, this allows the system to use compression rates as in a single SoPEC A4 architecture, but with the increased page size of A3.

4.2.5 A3 Duplex with 4 SoPEC devices

[0041] In Figure 5 a four SoPEC system is shown. It contains 2 A3 linking printheads, one for each side of an A3 page. Each printhead contain 16 printhead ICs, each SoPEC controls 8 printhead ICs. SoPEC #0 is the ISCMaster with the other SoPECs as ISCSlaves. Note that all 3 USB Host ports on SoPEC #0 are used to communicate with the 3 ISCSlave SoPECs. In total, the system contains 8Mbytes of compressed page store (2Mbytes per SoPEC), so the increased page size does not degrade the system print quality, from that of an A4 simplex printer. The ISCMaster receives all the compressed page data for all SoPECs and re-distributes the compressed data over the local USB bus to the ISCSlaves. This configuration could print 30 double-sided A3 sheets per minute.

4.2.6 SoPEC DRAM storage solution: A4 Simplex with 1 printing SoPEC and 1 memory SoPEC

[0042] Extra SoPECs can be used for DRAM storage e.g. in Figure 6 an A4 simplex printer can be built with a single extra SoPEC used for DRAM storage. The DRAM SoPEC can provide guaranteed bandwidth delivery of data to the printing SoPEC. SoPEC configurations can have multiple extra SoPECs used for DRAM storage.

4.2.7 Non-USB connection to Host PC

[0043] Figure 7 shows a configuration in which the connection from the host PC to the printer is an ethernet network, rather than USB. In this case, one of the USB Host ports on SoPEC interfaces to a external device that provide ethernet-to-USB bridging. Note that some networking software support in the bridging device might be required in this configuration. A Flash RAM will be required in such a system, to provide SoPEC with driver software for the Ethernet bridging function.

5 Printhead Misplacement Types

5.1 Printhead Construction

[0044] A linking printhead is constructed from linking printhead ICs , placed on a substrate containing ink supply holes. An A4 pagewidth printer used 11 linking printhead ICs. Each printhead is placed on the substrate with reference to positioning fidicuals on the substrate.

[0045] Figure 8 shows the arrangement of the printhead ICs (also known as segments) on a printhead. The join between two ICs is shown in detail. The left-most nozzles on each row are dropped by 10 line-pitches, to allow continuous printing across the join. Figure 8 also introduces some naming and co-ordinate conventions used throughout this document.

[0046] Figure 8 shows the anticipated first generation linking printhead nozzle arrangements, with 10 nozzle rows supporting five colours. The SoPEC compensation mechanisms are general enough to cover other nozzle arrangements.

5.2 Misplacement Types

[0047] Printheads ICs may be misplaced relative to their ideal position. This misplacement may include any combination of:

3. x offset
3. y offset
3. yaw (rotation around z)
3. pitch (rotation around y)
3. roll (rotation around z)

[0048] In some cases, the best visual results are achieved by considering relative misplacement between adjacent ICs, rather than absolute misplacement from the substrate. There are some practical limits to misplacement, in that a gross misplacement will stop the ink from flowing through the substrate to the ink channels on the chip.

[0049] Correcting for misplacement obviously requires the misplacement to be measured. In general this may be achieved directly by inspection of the printhead after assembly, or indirectly by scanning or examining a printed test pattern.

6 Misplacement compensation

6.1 X offset

[0050] SoPEC can compensate for misplacement of linking chips in the X-direction, but only snapped to the nearest dot. That is, a misplacement error of less than 0.5 dot-pitches or 7.9375 microns is not compensated for, a misplacement more that 0.5 dot-pitches but less than 1.5 dot-pitches is treated as a misplacement of 1 dot-pitch, etc.

[0051] Uncompensated X misplacement can result in three effects:

3. printed dots shifted from their correct position for the entire misplaced segment
3. missing dots in the overlap region between segments.
3. duplicated dots in the overlap region between segments.

[0052] SoPEC can correct for each of these three effects.

6.1.1 Correction for overall position in X

[0053] In preparing line data to be printed, SoPEC buffers in memory the dot data for a number of lines of the image to be printed. Compensation for misplacement generally involves changing the pattern in which this dot data is passed to the printhead ICs.

[0054] SoPEC uses separate buffers for the even and odd dots of each colour on each line, since they are printed by different printhead rows. So SoPEC's view of a line at this stage is as (up to) 12 rows of dots, rather than (up to) 6 colours. Nominally, the even dots for a line are printed by the lower of the two rows for that colour on the printhead, and the odd dots are printed by the upper row (see Figure 8). For the current linking printhead IC, there are 640 nozzles in row. Each row buffer for the full printhead would contain 640x11 dots per line to be printed, plus some padding if required.

[0055] In preparing the image, SoPEC can be programmed in the DWU module to precompensate for the fact that each row on the printhead IC is shifted left with respect to the row above. In this way the leftmost dot printed by each row for a colour is the same offset from the start of a row buffer. In fact the programming can support arbitrary shapes for the printhead IC.

[0056] SoPEC has independent registers in the LLU module for each segment that determine which dot of the prepared image is sent to the left-most nozzle of that segment. Up to 12 segments are supported. With no misplacement, SoPEC could be programmed to pass dots 0 to 639 in a row to segment 0, dots 640 to 1279 in a row to segment 1, etc.

[0057] If segment 1 was misplaced by 2 dot-pitches to the right, SoPEC could be adjusted to pass to dots 641 to 1280 of each row to segment 1 (remembering that each row of data consists entirely of either odd dots or even dots from a line, and that dot 1 on a row is printed two dot positions away from dot 0). This means the dots are printed in the correct position overall. This adjustment is based on the absolute placement of each printhead 1C. Dot 640 is not printed at all, since there is no nozzle in that position on the printhead (see Section 6.1.2 for more detail on compensation for missing dots).

[0058] A misplacement of an odd number of dot-pitches is more problematic, because it means that the odd dots from the line now need to be printed by the lower row of a colour pair, and the even dots by the upper row of a colour pair on the printhead segment. Further, swapping the odd and even buffers interferes with the precompensation. This results in the position of the first dot to be sent to a segment being different for odd and even rows of the segment. SoPEC addresses this by having independent registers in the LLU to specify the first dot for the odd and even rows of each segment, i.e. 2 x 12 registers. A further register bit determines whether dot data for odd and even rows should be swapped on a segment by segment basis.

6.1.2 Correcting for duplicate and missing dots

[0059] Figure 9 shows the detailed alignment of dots at the join between two printhead ICs, for various cases of misplacement, for a single colour.

[0060] The effects at the join depend on the relative misplacement of the two segments. In the ideal case with no misplacement, the last 3 nozzles of upper row of the segment N interleave with the first three nozzles of the lower row of segment N+1, giving a single nozzle (and so a single printed dot) at each dot-pitch.

[0061] When segment N+1 is misplaced to the right relative to segment N (a positive relative offset in X), there are some dot positions without a nozzle, i.e. missing dots. For positive offsets of an odd number of dot-pitches, there may also be some dot positions with two nozzles, i.e. duplicated dots. Negative relative offsets in X of segment N+1 with respect to segment N are less likely, since they would usually result in a collision of the printhead ICs, however they are possible in combination with an offset in Y. A negative offset will always cause duplicated dots, and will cause missing dots in some cases. Note that the placement and tolerances can be deliberately skewed to the right in the manufacturing step to avoid negative offsets.

[0062] Where two nozzles occupy the same dot position, the corrections described in Section 6.1.1 will result in SoPEC reading the same dot data from the row buffer for both nozzles. To avoid printing this data twice SoPEC has two registers per segment in the LLU that specify a number (up to 3) of dots to suppress at the start of each row, one register applying to even dot rows, one to odd dot rows.

[0063] SoPEC compensates for missing dots by add the missing nozzle position to its dead nozzle map. This tells the dead nozzle compensation logic in the DNC module to distribute the data from that position into the surrounding nozzles, before preparing the row buffers to be printed.

6.2 Y Offset

[0064] SoPEC can compensate for misplacement of printhead ICs in the Y-direction, but only snapped to the nearest 0.1 of a line. Assuming a line-pitch of 15.875 microns, if an IC is misplaced in Y by 0 microns, SoPEC can print perfectly in Y. If an IC is misplaced by 1.5875 microns in Y, then we can print perfectly. If an IC is misplaced in Y by 3.175 microns, we can print perfectly. But if an IC is misplaced by 3 microns, this is recorded as a misplacement of 3.175 microns (snapping to the nearest 0.1 of a line), and resulting in a Y error of 0.175 microns (most likely an imperceptible error).

[0065] Uncompensated Y misplacement results in all the dots for the misplaced segment being printed in the wrong position on the page.

[0066] SoPEC's compensation for Y misplacement uses two mechanism, one to address whole line-pitch misplacement, and another to address fractional line-pitch misplacement. These mechanisms can be applied together, to compensate for arbitrary misplacements to the nearest 0.1 of a line.

6.2.1 Compensating for whole line-pitch misplacement

[0067] Section 6.1 described the buffers used to hold dot data to be printed for each row. These buffers contain dot data for multiple lines of the image to be printed. Due to the physical separation of nozzle rows on a printhead IC, at any time different rows are printing data from different lines of the image.

[0068] For a printhead on which all ICs are ideally placed, row 0 of each segment is printing data from the line N of the image, row 1 of each segment is printing data from row N-M of the image etc. where N is the separation of rows 0 and 1 on the printhead. Separate SoPEC registers in the LLU for each row specify the designed row separations on the printhead, so that SoPEC keeps track of the "current" image line being printed by each row.

[0069] If one segment is misplaced by one whole line-pitch, SoPEC can compensate by adjusting the line of the image being sent to each row of that segment. This is achieved by adding an extra offset on the row buffer address used for that segment, for each row buffer. This offset causes SoPEC to provide the dot data to each row of that segment from one line further ahead in the image than the dot data provided to the same row on the other segments. For example, when the correctly placed segments are printing line N of an image with row 0, line N-M of the image with row 1, etc, then the misplaced segment is printing line N+1 of the image with row 0, line N-M+1 of the image with row 1, etc.

[0070] SoPEC has one register per segment to specify this whole line-pitch offset. The offset can be multiple line-pitches, compensating for multiple lines of misplacement. Note that the offset can only be in the forward direction, corresponding to a negative Y offset. This means the initial setup of SoPEC must be based on the highest (most positive) Y-axis segment placement, and the offsets for other segments calculated from this baseline. Compensating for Y displacement requires extra lines of dot data buffering in SoPEC, equal to the maximum relative Y offset (in line-pitches) between any two segments on the printhead. For each misplaced segment, each line of misplacement requires approximately 640x10 or 6400 extra bits of memory.

6.2.2 Compensation for fractional line-pitch misplacement

[0071] Compensation for fractional line-pitch displacement of a segment is achieved by a combination of SoPEC and printhead IC fire logic.

[0072] The nozzle rows in the printhead are positioned by design with vertical spacings in line-pitches that have a integer and fractional component. The fractional components are expressed relative to row zero, and are always some multiple of 0.1 of a line-pitch. The rows are fired sequentially in a given order, and the fractional component of the row spacing matches the distance the paper will move between one row firing and the next. Figure 10 shows the row position and firing order on the current implementation of the printhead IC. Looking at the first two rows, the paper moves by 0.5 of a line-pitch between the row 0 (fired first) and row 1 (fired sixth). is supplied with dot data from a line 3 lines before the data supplied to row 0. This data ends up on the paper exactly 3 line-pitches apart, as required.

[0073] If one printhead IC is vertically misplaced by a non-integer number of line-pitches, row 0 of that segment no longer aligns to row 0 of other segments. However, to the nearest 0.1 of a line, there is one row on the misplaced segment that is an integer number of line-pitches away from row 0 of the ideally placed segments. If this row is fired at the same time as row 0 of the other segments, and it is supplied with dot data from the correct line, then its dots will line up with the dots from row 0 of the other segments, to within a 0.1 of a line-pitch. Subsequent rows on the misplaced printhead can then be fired in their usual order, wrapping back to row 0 after row 9. This firing order results in each row firing at the same time as the rows on the other printheads closest to an integer number of line-pitches away.

[0074] Figure 11 shows an example, in which the misplaced segment is offset by 0.3 of a line-pitch. In this case, row 5 of the misplaced segment is exactly 24.0 line-pitches from row 0 of the ideal segment. Therefore row 5 is fired first on the misplaced segment, followed by row 7, 9, 0 etc. as shown. Each row is fired at the same time as the a row on the ideal segment that is an integer number of lines away. This selection of the start row of the firing sequence is controlled by a register in each printhead IC.

[0075] SoPEC's role in the compensation for fractional line-pitch misplacement is to supply the correct dot data for each row. Looking at Figure 11, we can see that to print correct, row 5 on the misplaced printhead needs dot data from a line 24 lines earlier in the image than the data supplied to row 0. On the ideal printhead, row 5 needs dot data from a line 23 lines earlier in the image than the data supplied to row 0. In general, when a non-default start row is used for a segment, some rows for that segment need their data to be offset by one line, relative to the data they would receive for a default start row. SoPEC has a register in LLU for each row of each segment, that specifies whether to apply a one line offset when fetching data for that row of that segment.

6.3 Roll (rotation around X)

[0076] This kind of erroneous rotational displacement means that all the nozzles will end up pointing further up the page in Y or further down the page in Y. The effect is the same as a Y misplacement, except there is a different Y effect for each media thickness (since the amount of misplacement depends on the distance the ink has to travel).

[0077] In some cases, it may be that the media thickness makes no effective visual difference to the outcome, and this form of misplacement can simply be incorporated into the Y misplacement compensation. If the media thickness does make a difference which can be characterised, then the Y misplacement programming can be adjusted for each print, based on the media thickness.

[0078] It will be appreciated that correction for roll is particularly of interest where more than one printhead module is used to form a printhead, since it is the discontinuities between strips printed by adjacent modules that are most objectionable in this context.

6.4 Pitch (rotation around Y)

[0079] In this rotation, one end of the IC is further into the substrate than the other end. This means that the printing on the page will be dots further apart at the end that is further away from the media (i.e. less optical density), and dots will be closer together at the end that is closest to the media (more optical density) with a linear fade of the effect from one extreme to the other. Whether this produces any kind of visual artifact is unknown, but it is not compensated for in SoPEC.

6.5 Yaw (rotation around Z)

[0080] This kind of erroneous rotational displacement means that the nozzles at one end of a IC will print further down the page in Y than the other end of the IC. There may also be a slight increase in optical density depending on the rotation amount.

[0081] SoPEC can compensate for this by providing first order continuity, although not second order continuity in the preferred embodiment. First order continuity (in which the Y position of adjacent line ends is matched) is achieved using the Y offset compensation mechanism, but considering relative rather than absolute misplacement. Second order continuity (in which the slope of the lines in adjacent print modules is at least partially equalised) can be effected by applying a Y offset compensation on a per pixel basis. Whilst one skilled in the art will have little difficulty deriving the timing difference that enables such compensation, SoPEC does not compensate for it and so it is not described here in detail.

[0082] Figure 12 shows an example where printhead IC number 4 is be placed with yaw, is shown in Figure 12, while all other ICs on the printhead are perfectly placed. The effect of yaw is that the left end of segment 4 of the printhead has an apparent Y offset of -1 line-pitch relative to segment 3, while the right end of segment 4 has an apparent Y offset of 1 line-pitch relative to segment 5.

[0083] To provide first-order continuity in this example, the registers on SoPEC would be programmed such that segments 0 to 3 have a Y offset of 0, segment 4 has a Y offset of - 1, and segments 5 and above have Y offset of -2. Note that the Y offsets accumulate in this example - even though segment 5 is perfect aligned to segment 3, they have different Y offsets programmed.

[0084] It will be appreciated that some compensation is better than none, and it is not necessary in all cases to perfectly correct for roll and/or yaw. Partial compensation may be adequate depending upon the particular application. As with roll, yaw correction is particularly applicable to multi-module printheads, but can also be applied in single module printheads.

7. Printhead Requirements

7.1 NUMBER OF COLORS

[0085] The printhead will be designed for 5 colors. At present the intended use is:

(i) cyan
(ii) magenta
(iii) yellow
(iv) black
(v) infra-red

[0086] However the design methodology must be capable of targeting a number other than 5 should the actual number of colors change. If it does change, it would be to 6 (with fixative being added) or to 4 (with infra-red being dropped).

[0087] The printhead chip does not assume any particular ordering of the 5 colour channels.

7.2 NUMBER OF NOZZLES

[0088] The printhead will contain 1280 nozzles of each color - 640 nozzles on one row firing even dots, and 640 nozzles on another row firing odd dots. This means 11 linking printheads are required to assemble an A4/Letter printhead.

[0089] However the design methodology must be capable of targeting a number other than 1280 should the actual number of nozzles per color change. Any different length may need to be a multiple of 32 or 64 to allow for ink channel routing.

7.3 NOZZLE SPACING

[0090] The printhead will target true 1600 dpi printing. This means ink drops must land on the page separated by a distance of 15.875 microns.

[0091] The 15.875 micron inter-dot distance coupled with mems requirements mean that the **horizontal** distance between two adjacent nozzles on a single row (e.g. firing even dots) will be 31.75 microns.

[0092] All 640 dots in an odd or even colour row are exactly aligned **vertically**. Rows are fired sequentially, so a complete row is fired in small fraction (nominally one tenth) of a line time, with individual nozzle firing distributed within this row time. As a result dots can end up on the paper with a vertical misplacement of up to one tenth of the dot pitch. This is considered acceptable.

[0093] The vertical distance between rows is adjusted based on the row firing order. Firing can start with any row, and then follows a fixed rotation. Figure 13 shows the default row firing order from 1 to 10, starting at the top even row. Rows are separated by an exact number of dot lines, plus a fraction of a dot line corresponding to the distance the paper will move between row firing times. This allows exact dot-on-dot printing for each colour. The starting row can be varied to correct for vertical misalignment between chips, to the nearest 0.1 pixels. SoPEC appropriate delays each row's data to allow for the spacing and firing order

[0094] An additional constraint is that the odd and even rows for given colour must be placed close enough together to allow them to share an ink channel. This results in the vertical spacing shown in Figure 364, where L represents one dot pitch.

7.4 LINKING THE CHIPS

[0095] Multiple identical printhead chips must be capable of being linked together to form an effectively horizontal assembled printhead.

[0096] Although there are several possible internal arrangements, construction and assembly tolerance issues have made an internal arrangement of a dropped triangle (ie a set of rows) of nozzles within a series of rows of nozzles, as shown in Figure 14. These printheads can be linked together as shown in Figure 15.

[0097] Compensation for the triangle is preferably performed in the printhead, but if the storage requirements are too large, the triangle compensation can occur in SoPEC. However, if the compensation is performed in SoPEC, it is required in the present embodiment that there be an even number of nozzles on each side of the triangle.

[0098] It will be appreciated that the triangle disposed adjacent one end of the chip provides the minimum on-printhead storage requirements. However, where storage requirements are less critical, other shapes can be used. For example, the dropped rows can take the form of a trapezoid.

[0099] The join between adjacent heads has a 45° angle to the upper and lower chip edges. The joining edge will not be straight, but will have a sawtooth or similar profile. The nominal spacing between tiles is 10 microns (measured perpendicular to the edge). SoPEC can be used to compensate for both horizontal and vertical misalignments of the print heads, at some cost to memory and/or print quality.

[0100] Note also that paper movement is fixed for this particular design.

7.5 PRINT RATE

[0101] A print rate of 60 A4/Letter pages per minute is possible. The printhead will assume the following:

3. page length = 297mm (A4 is longest page length)

3. an inter-page gap of 60mm or less (current best estimate is more like 15 +/- 5mm This implies a line rate of 22,500 lines per second. Note that if the page gap is not to be considered in page rate calculations, then a 20KHz line rate is sufficient.

[0102] Assuming the page gap is required, the printhead must be capable of receiving the data for an entire line during the line time. i.e. 5 colors 1280 dots 22,500 lines = 144MHz or better (173MHz for 6 colours).

7.6 PINS

[0103] An overall requirement is to minimize the number of pins.

[0104] Pin count is driven primarily by the number of supply and ground pins for Vpos. There is a lower limit for this number based on average current and electromigration rules. There is also a significant routing area impact from using fewer supply pads.

[0105] In summary a 200nJ ejection energy implies roughly 12.5W average consumption for 100% ink coverage, or 2.5W per chip from a 5V supply. This would mandate a minimum of 20 Vpos/Gnd pairs. However increasing this to around 40 pairs might save approximately 100 microns from the chip height, due to easier routing.

[0106] At this stage the print head is assuming 40 Vpos/Gnd pairs, plus 11 Vdd (3.3V) pins, plus 6 signal pins, for a total of 97 pins per chip.

7.7 INK SUPPLY HOLE

[0107] At the CMOS level, the ink supply hole for each nozzle is defined by a metal seal ring in the shape of rectangle (with square corners), measuring 11 microns horizontally by 26 microns vertically. The centre of each ink supply hole is directly under the centre of the MEMs nozzle, i.e. the ink supply hole horizontal and vertical spacing is same as corresponding nozzle spacing.

7.8 ESD

[0108] The printhead will most likely be inserted into a print cartridge for user-insertion into the printer, similar to the way a laser-printer toner cartridge is inserted into a laser printer.

[0109] In a home/office environment, ESD discharges up to 15kV may occur during handling. It is not feasible to provide protection against such discharges as part of the chip, so some kind of shielding will be needed during handling.

[0110] The printhead chip itself will target MIL-STD-883 class 1 (2kV human body model), which is appropriate for assembly and test in a an ESD-controlled environment.

7.9 EMI

[0111] There is no specific requirement on EMI at this time, other than to minimize emissions where possible.

7.10 HOT PLUG / UNPLUG

[0112] Cartridge (and hence printhead) removal may be required for replacement of the cartridge or because of a paper jam.

[0113] There is no requirement on the printhead to withstand a hot plug/unplug situation. This will be taken care of by the cradle and/or cartridge electromechanics.

7.11 POWER SEQUENCING

[0114] The printhead does not have a particular requirement for sequencing of the 3.3V and 5V supplies. However there is a requirement to held reset asserted (low) as power is applied.

7.12 POWER-ON RESET

[0115] Will be supplied to the printhead. There is no requirement for Power-on-Reset circuitry inside the printhead.

7.13 OUTPUT VOLTAGE RANGE

[0116] Any output pins (typically going to SoPEC) will drive at 3.3VDD +- 5%.

7.14 TEMPERATURE RANGE

[0117] The print head CMOS will be verified for operation over a range of -10C to 11 10C.

7.15 RELIABILITY AND LIFETIME

[0118] The print head CMOS will target a lifetime of at least 10 billion ejections per nozzle.

8 Physical Overview

[0119] The SRM043 is a CMOS and MEMS integrated chip. The MEMS structures/nozzles can eject ink which has passed through the substrate of the CMOS via small etched holes.

[0120] The SRM043 has nozzles arranged to create a accurately placed 1600 dots per inch printout. The SRM043 has 5 colours, 1280 nozzles per colour.

[0121] The SRM043 is designed to link to a similar SRM043 with perfect alignment so the printed image has no artifacts across the join between the two chips.

[0122] SRM043 contains 10 rows of nozzles, arranged as upper and lower row pairs of 5 different inks. The paired rows share a common ink channel at the back of the die. The nozzles in one of the paired rows are horizontally spaced 2 dot pitches apart, and are offset relative to each other.

8.1 COLOUR ARRANGEMENT

[0123] 1600 dpi has a dot pitch of *DP* 15.875 m. The MEMS print nozzle unit cell is *2DP* wide by *5DP* high (31.75 m x 79.375 m). To achieve 1600 dpi per colour, 2 horizontal rows of (1280/2) nozzles are placed with a horizontal offset of *5DP* (2.5 cells). Vertical offset is 3.5DP between the two rows of the same colour and 10.1DP between rows of different colour. This slope continues between colours and results in a print area which is a trapezoid as shown in Figure 16.

[0124] Within a row, the nozzles are perfectly aligned vertically.

8.2 LINKING NOZZLE ARRANGEMENT

[0125] For ink sealing reasons a large area of silicon beyond the end nozzles in each row is required on the base of the die, near where the chip links to the next chip. To do this the first 4*Row#+4 -2*(Row#mod2) nozzles from each row are vertical shifted down DP. Data for the nozzles in the triangle must be delayed by 10 line times to match the triangle vertical offset.. The appropriate number of data bits at the start of each row are put into a FIFO. Data from the FIFO's output is used instead. The rest of the data for the row bypasses the FIFO.

9 Block Diagram and Overview of CMOS Modules

[0126] Figure 19 shows the top levels of the block diagram and by extension the top wrapper netlist for the printhead.

[0127] The modules comprising the linking printhead CMOS are:

9.1 CORE

[0128] The core contains an array of unit cells and the column shift register (columnSR).

[0129] The Unit Cell is the base structure of the printhead, consisting of one bit of the row data shift register, a latch to double buffer the data, the MEMS ink firing mechanism, a large transistor to drive the MEMS and some gates to enable that transistor at the correct time.

[0130] The column shift register is at the bottom of the core unit cell array. It is used to generate timing for unit cell firing, in conjunction with the fpg.

9.2 TRIANGLE DELAY COMPENSATION (TDC)

[0131] The TDC module handles the loading of data into row shift regsiters of the core.

[0132] The dropped triangle at the left hand end of the core prints 10 lines lower on the page than the bulk of each row. This implies data has to be delayed by 10 line times before ink ejection. To minimize overhead on the print controller, and to make the interface cleaner, that delay is provided on chip.

[0133] The TDC block connects to a fifo used to store the data to be delayed, and routes the first few nozzle data samples in a particular row with data through the fifo. All subsequent data is passed straight through to the row shift registers.

[0134] The TDC also serializes 8 bit wide data at the symbol rate of 28.8MHz to 2 bit nibbles at a 144MHz rate, routes that data to all row shift registers, and synchronously generates gated clocks for the addressed row shift register.

9.3 FPG

[0135] The Fire and Profile Generator controls the firing sequence of the nozzles on a row and column basis, and the width of the firing pulses applied to to each actuator.

[0136] It produces timed profile pulses for each row of the core. It also generates clock and data to drive the ColumnSR. The column enables from the ColumnSR, the row profile, and the data within the core are all and'ed together to fire the unit cell actuators and hence eject ink.

[0137] The FPG sequences the firing to produce accurate dot placement, compensating for printhead position and generates correct width profiles.

9.4 DEX

[0138] The Data EXtractor converts the input data stream into byte-wide command and data symbols to the CU. It interfaces with a full-custom Datamux to sample data presented to the chip at the optimum eye. This data is then descrambled, symbols are aligned and deserialized, and then decoded. Data and symbol type is passed to the CU.

9.5 CU

[0139] The Command Unit contains most of the control registers. It is responsible for implementing the command protocol, and routes control and data and clocks to the rest of the chip as appropriate. The CU also contains all BIST functionality.

[0140] The CU synchronizes reset_n for the rest of the chip. Reset is removed synchronously, but is applied to flip flops on the async clear pin. Fire enable is overridden with an asynchronous reset signal.

9.6 IO

[0141] The chip has high speed clock and data LVDS pads connected to the DEX module.

[0142] There is a Reset_n input and a modal tristate/open drain output managed by the CU.

[0143] There are also a number of ground pads, VDD pads and also VPOS pads for the unit cell.

[0144] The design should have no power sequencing requirements, but does require reset_n to be asserted at power on.

[0145] Lack of power sequencing requires that the ESD protection in the pads be to ground, there cannot be diodes between the VPOS and VDD rails.

[0146] Similarly the level translator in the unit cell must ensure that the PMOS switching transistor is off in the event VPOS is up before VDD.

9.7 NORMAL OPERATION

[0147] The normal operation of the linking printhead is
reset the head
program registers to control the firing sequence and parameters
load data for a single print line into (up to) 10 rows of the printhead send a FIRE command, which latches the loaded data, and begins a fire cycle while the fire cycle is in progress, load data for the next print line if the page is not finished, goto 4.

[0148] Note the spacing of FIRE commands determines the printing speed (in lines/second). The printhead would normally be set up so that a fire cycle takes all of the time available between FIRE commands.

10. Detailed Description of TDC Module

10.1 TDC 10

[0149]

Table 1. TDC 10

Signal	Drn	to/from	Description
di[7:0]	in	from: CU	8 bit row data, at symbol (clk28) rate
data_valid	in	from: CU	enable for data, in clk28 domain
clk	in	from: IO	288MHz clock
phi9	in	from: DEX	synchronizing clk signal
tdc_bypass	in	from: CU	disable triangle delay compensation
ld_n	in	from: CU	initiate fire cycle
do[1;0]	out	to: core	output data to core row shift registers
rclk[9:0]	out	to: core	core shift register row clocks. 144MHz gated clocks, no more than one running at a time.
row[3:0]	in	from: CU	core row to write to
newrow	in	from: CU	the core row has changed, recalculate.
fifo_di[1 :0]	out	to: TDC_FIFO	first up delayed data
fifo_do[1:0]	in	from: TDC_fifo	delayed data from fifo
fifo_clk	out	to: TDC_fifo	fifo clock. 144MHz gated clock, aligned to rclks
Single_r	in	from: CU	generates a single rclk event when asserted,
clk			used for core readback.

10.2 Functionality

[0150] The TDC receives row data from the CU, partially serializes it, and writes it to the currently addressed printhead row. It also strips the required number of bits from the beginning of the row and stores them in the TDC_fifo, replacing them with bits shifted out of the TDC_fifo. This occurs transparently to the master SoPEC.

[0151] The TDC generates a local symbol phase clock using phi9. This clock phase information, together with the data_valid level, is used to generate fifo and row clocks. These clocks are timed as shown in Figure 21. The precise number of fifo clocks per row is shown in Table 3.

[0152] The CU indicates when the current addressed row changes. That row is mapped to get the number of bits to pass through the fifo, and also whether the number of fifo bits is odd. [The current FIFO is never odd, but this has not always been the case so the logic remains in the RTL] A counter is loaded with the total number of required clocks, and then allowed to count down. When it reaches terminal count, a done flag is set, This flag is used to indicate whether row data is delayed through the fifo, or passed directly to the core. There is a single done flag, so a row can only be addressed once per fire cycle.

[0153] If the number of bits to delay is odd, and the counter has reached terminal count, then one bit for the core is taken from the fifo and one bit from the current presented byte. The fifo bit used is always on fifo_do[0]. fifo_do[1] is discarded in this case.

[0154] A tdc_bypass bit always causes data to bypass the fifo, and pass directly to the core. This mode may be used for print test, for nozzle unclogging and potentially if SoPEC was to be used to compensate for the triangle delay.

[0155] This design allows the core to be randomly addressed if required,. All lines on a page must be written in the same row order. Once a row has started writing, it must be completed. At least enough symbols to fill the TDC fifo fragment must be sent for every row for every line. If fewer than 80 but at least the number shown in Table 3 centre column are sent, the TDC will work correctly but under-run errors will be reported by CU.

[0156] Not withstanding the above, if the single_rclk input is asserted, then a rclk[] for the row currently pointed at will be generated. This rclk may be asserted in the next odd clk phase. This rclk is a single cycle of clk in width, and there is only one. There is no control over the two bits written to core in this mode.

10.3 TDC FIFO

10.3.1 TDC FIFO IO

[0157]

Table 2. TDC FIFO IO

Signal	Drn	to/from	Description
fifo_di[1 :0]	in	from: tdc	Fifo data in
fifo_do[1:0]	out	to: tdc	Fifo data out
fifo_clk	in	from: tdc	fifo clock at 144MHz. This clock is generated as a burst clock in the tdc module.

10.3.2 TDC FIFO Functionality

[0158] To allow the printheads to abut seamlessly there is a section at the far left of the core where a triangular group of nozzles, some from each row, is shifted down. This increases the linear distance between consecutive nozzles in the same logical row across the join, allowing simpler ink sealing between the printhead and the ink distribution system. It will be appreciated that the size and shape of the dropped rows is arbitrary, but that making them triangular and minimal in size has the desirable impact of reducing the amount of memory requird to hold the data in the dropped rows.

[0159] The number of nozzles in the dropped triangle differs for each row and is shown in Table 3. These nozzles will fire 10 fire cycles after the rest of the row, resulting in ink being aligned on paper with the main part of the row. To facilitate this the bits to be delayed are written to a fifo called tdc_fifo. This delays those bits by 10 rows.

[0160] As the core shift registers are intrinsically 2 bits wide, the fifo is made 2 bits also., and is clocked at the same rate as the row shift regsiters, 144MHz. We have chosen to clock both fifo rows with a common clock for implementation reasons. This requires us to add a few extra locations to the fifo if the number of fifo location is odd for a particular row.

[0161] 320 row clocks are generated to load a complete core row. The fifo is clocked for a variable number of clocks at the start of a row, as shown in Table 3.

Table 3. Triangle rows

Row	Nozzles in drop triangle	FIFO clocks at start of row
0	4	2
1	6	3
2	12	6
3	14	7
4	20	10
5	22	11

(continued)

Row	Nozzles in drop triangle	FIFO clocks at start of row
6	28	14
7	30	15
8	36	18
9	38	19
Subtota 1	210	

[0162] The triangle is dropped 10 rows, so there are 2100 flip flops required in he TDC_fifo. This must be shaped as 2x1050.

10.3.4 TDC FIFO Implementation

[0163] The TDC_fifo is implemented as a hard macro to minimize area requirements.

[0164] A verilog netlist is written using instantiated custom-made flip flops. The flipflop used is the same as that used in the shift register. It is optimized for size, being around one third of a standard TSMC flipflop in size. It has limited drive and requires both clock and clock_bar to operate.

[0165] The design uses a repeating set of 8 columns, where data weaves up and down, one pair to the left and one pair to the right. These two columns are connected at the lower left to form a 2 bit wide shift register. Inputs and outputs are all at the lower right hand corner.

[0166] This implementation yields a synchronous IO referenced to a local clock, and also allows regular clock buffering along the die. Spice is used to verify setup and hold times are met everywhere.

[0167] The gated clock is chosen for power reasons. This clock is generated in the TDC using a 288MHz clock. The TDC fifo can stream data at 144MHz and has a delay of 1050 (for a 10 row printhead) clocks. The fifo is rising edge clock triggered.

10.3.5 Timing

[0168] The TDC fifo has a latency of 1050 clocks.

11. Printing fewer than the full number of channels available on the printhead

[0169] It is possible to use SoPEC to send dot data to a printhead that is using less than its full complement of rows. For example, it is possible that the fixative, IR and black channels will be omitted in a low end, low cost printer. Rather than design a new printhead having only three channels, it is possible to select which channels are active in a printhead with a larger number of channels (such as the presently preferred channel version). It may be desirable to use a printhead which has one or more defective nozzles in up to three rows as a printhead (or printhead module) in a three color printer.

[0170] It would be disadvantageous to have to load empty data into each empty channel, so it is preferable to allow one or more rows to be disabled in the printhead.

[0171] The printhead already has a register that allows each row to be individually enabled or disabled (register ENABLE at address 0). Currently all this does is suppress firing for a non-enabled row.

[0172] To avoid SoPEC needing to send blank data for the unused rows, the functionality of these bits is extended to:

1. skip over disabled rows when DATA_NEXT register is written;

2. force dummy bits into the TDC FIFO for a disabled rows, corresponding to the number of nozzles in the dropped triangle section for that row. These dummy bits are written immediately following the first row write to the fifo following a fire command.

[0173] Using this arrangement, it is possible to operate a 6 color printhead as a 1 to 6 color printhead, depending upon which mode is set. The mode can be set by the printer controller (SoPEC); once set, SoPEC need only send dot data for the active channels of the printhead.

Claims

1. A printhead module including at least one row of printhead nozzles, at least one row including at least one displaced row portion, the displacement of the row portion including a component in a direction normal to that of a pagewidth to be printed.

2. A printhead module according to claim 1, wherein the displaced row portion is disposed adjacent one end of the printhead module.

3. A printhead module according to claim 1, including a plurality of the rows, wherein each of at least a plurality of the rows includes one of the displaced row portions.

4. A printhead module according to claim 3, wherein the displaced row portions of at least some of the rows are different in length than the displaced row portions of at least some of the other rows.

5. A printhead module according to claim 4, wherein each of the rows has a displaced row portion, and the sizes of the respective displaced row portions increase from row to row in the direction normal to that of the pagewidth to be printed.

6. A printhead module according to claim 5, wherein the dropped rows together comprise a generally trapezoidal shape, in plan.

7. A printhead module according to claim 5, wherein the dropped rows together comprise a generally triangular shape, in plan.

8. A printhead module according to any one of the preceding claims, wherein the printhead module is a printhead integrated circuit.

9. A printhead comprising a plurality of printhead modules, including at least one of the printhead modules according to any one of the preceding claims.

10. A printhead according to claim 9, wherein the printhead modules are the same shape and configuration as each other, and are arranged end to end across the intended print width.

11. A printhead according to claim 9 or claim 10, wherein the printhead modules together define a pagewidth printhead.

12. A printer controller for supplying data to a printhead module according to any one of claims 1 to 7, the printer controller being configured to control order and timing of the data supplied to the printhead such that the dropped row is compensated for during printing by the printhead module.

13. An inkjet printer comprising the printhead according to any one of claims 9 to 11 and the printer controller according to claim 12.

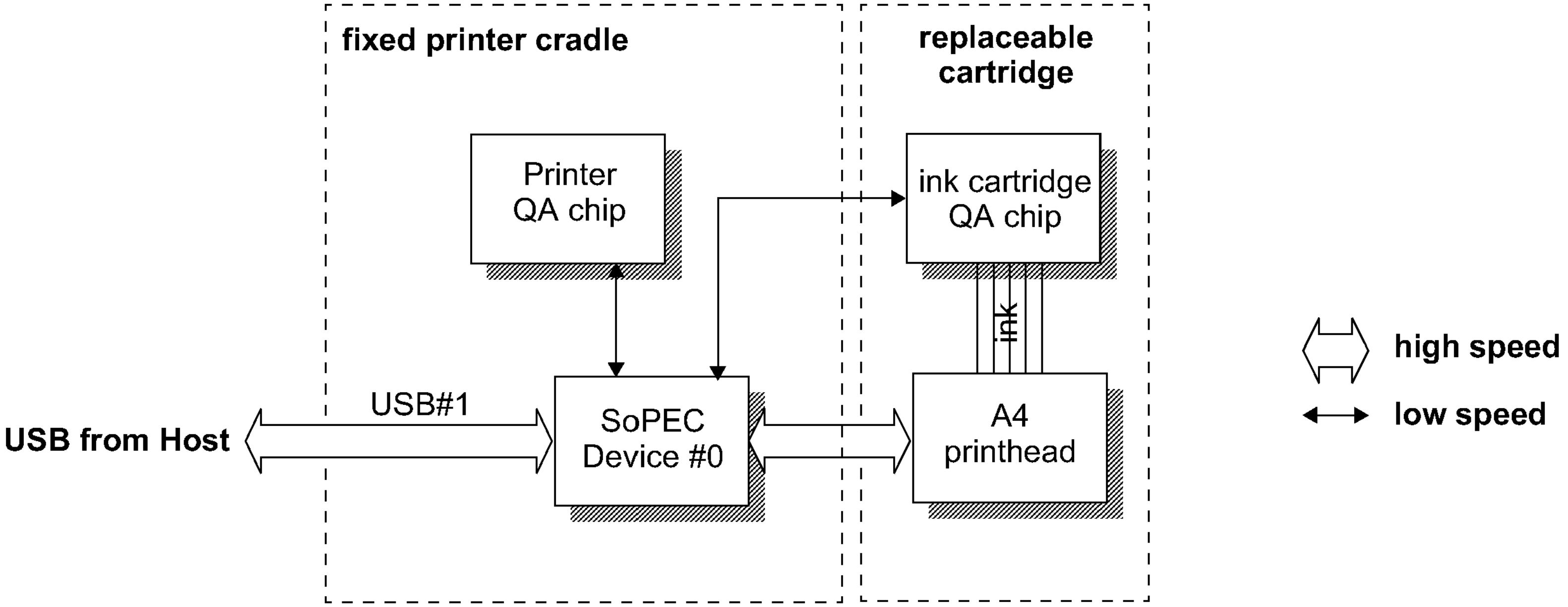
fixed printer cradle
replaceable cartridge
Printer QA chip
ink cartridge QA chip
ink
USB#1
USB from Host
SoPEC Device #0
A4 printhead
high speed
low speed

FIG. 1

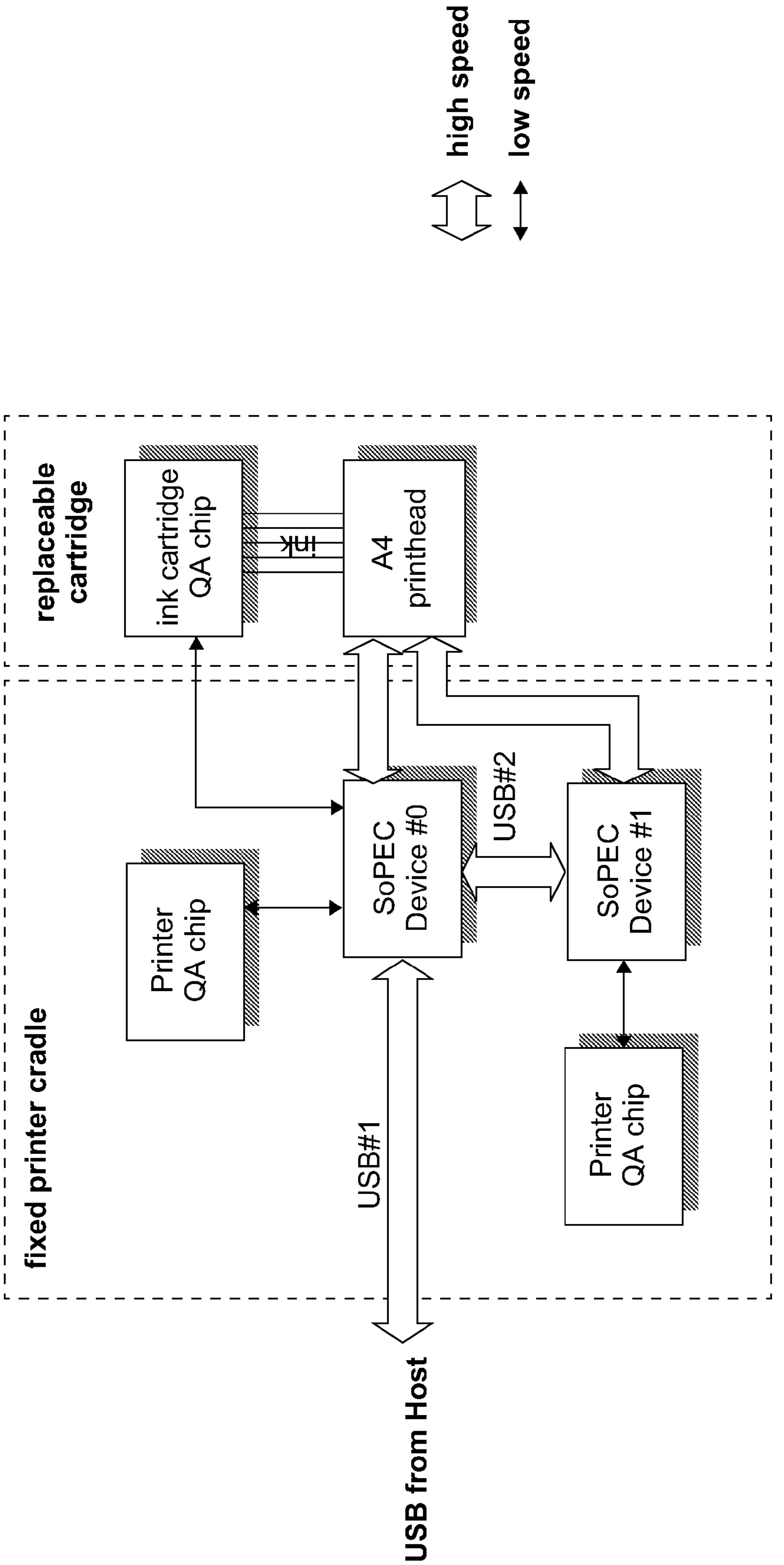
high speed
low speed
replaceable cartridge
ink cartridge QA chip
ink
A4 printhead
fixed printer cradle
Printer QA chip
SoPEC Device #0
USB#2
SoPEC Device #1
Printer QA chip
USB#1
USB from Host

FIG. 2

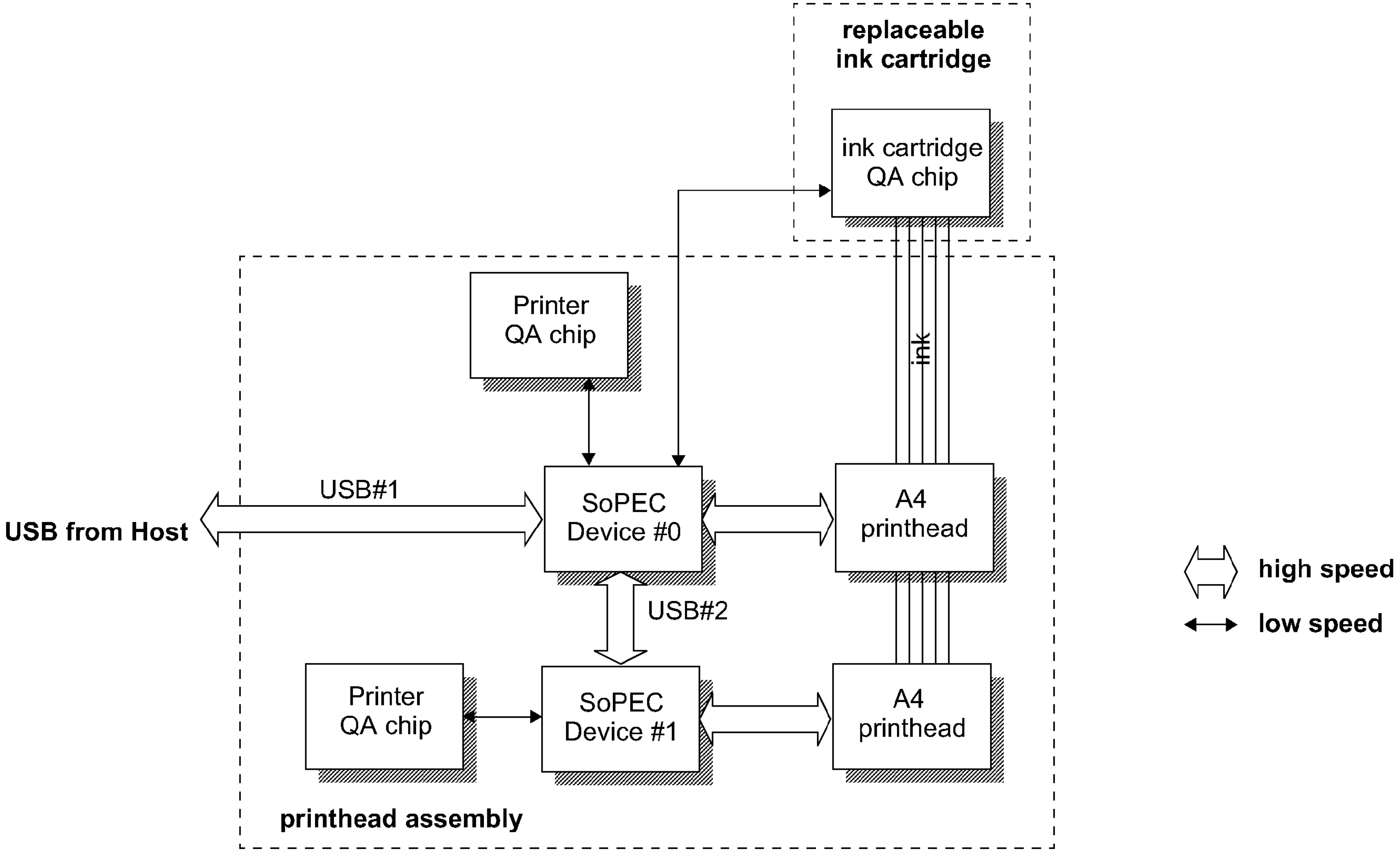
replaceable
ink cartridge
ink cartridge
QA chip
Printer
QA chip
ink
USB#1
USB from Host
SoPEC
Device #0
A4
printhead
USB#2
high speed
low speed
Printer
QA chip
SoPEC
Device #1
A4
printhead
printhead assembly

FIG. 3

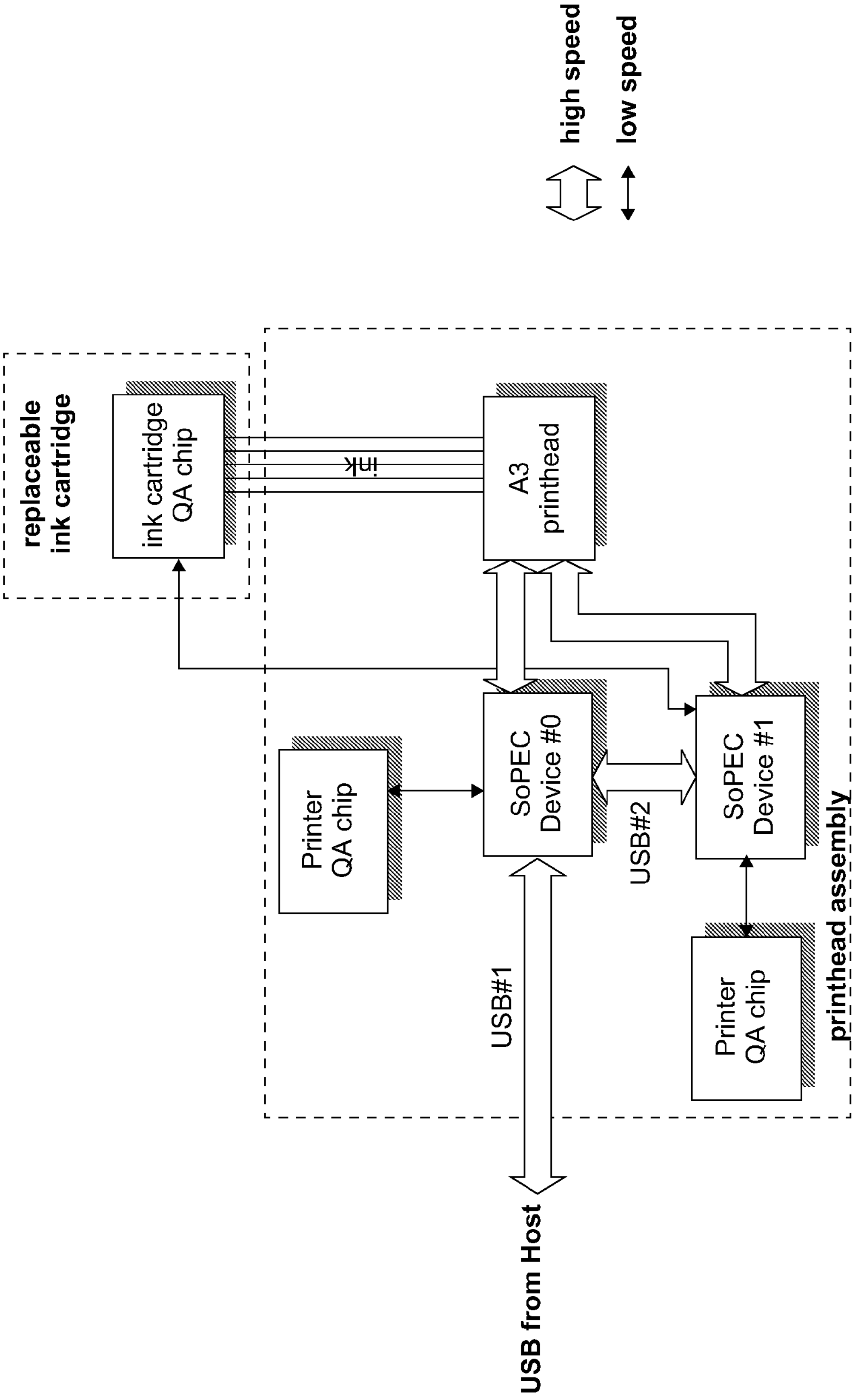
high speed
low speed
replaceable ink cartridge
ink cartridge QA chip
ink
A3 printhead
Printer QA chip
SoPEC Device #0
SoPEC Device #1
USB#2
USB#1
Printer QA chip
printhead assembly
USB from Host

FIG. 4

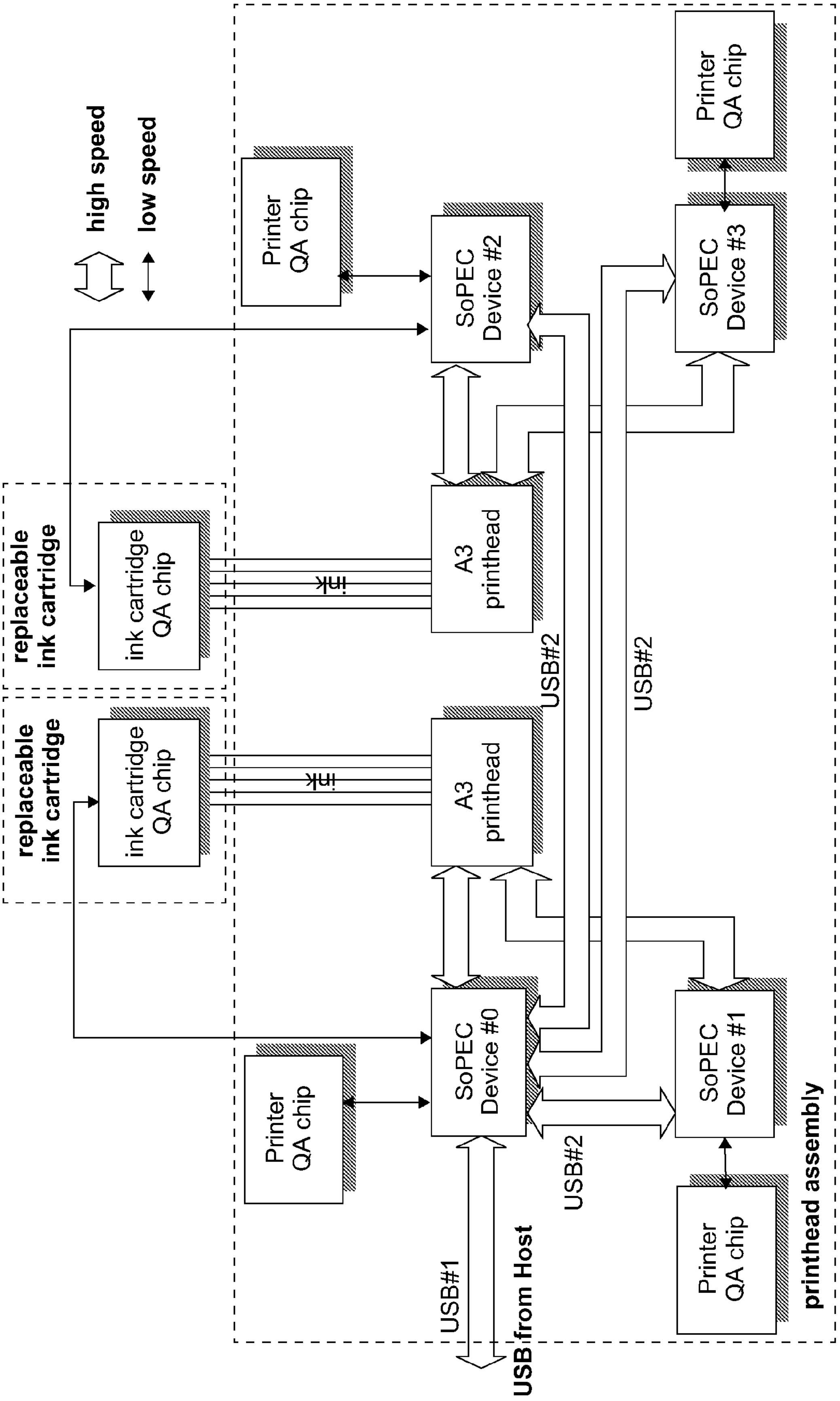
high speed
low speed
replaceable ink cartridge
ink cartridge QA chip
replaceable ink cartridge
ink cartridge QA chip
Printer QA chip
SoPEC Device #2
ink
A3 printhead
ink
A3 printhead
Printer QA chip
SoPEC Device #0
USB#1
USB from Host
USB#2
USB#2
USB#2
Printer QA chip
SoPEC Device #3
SoPEC Device #1
Printer QA chip
printhead assembly

FIG. 5

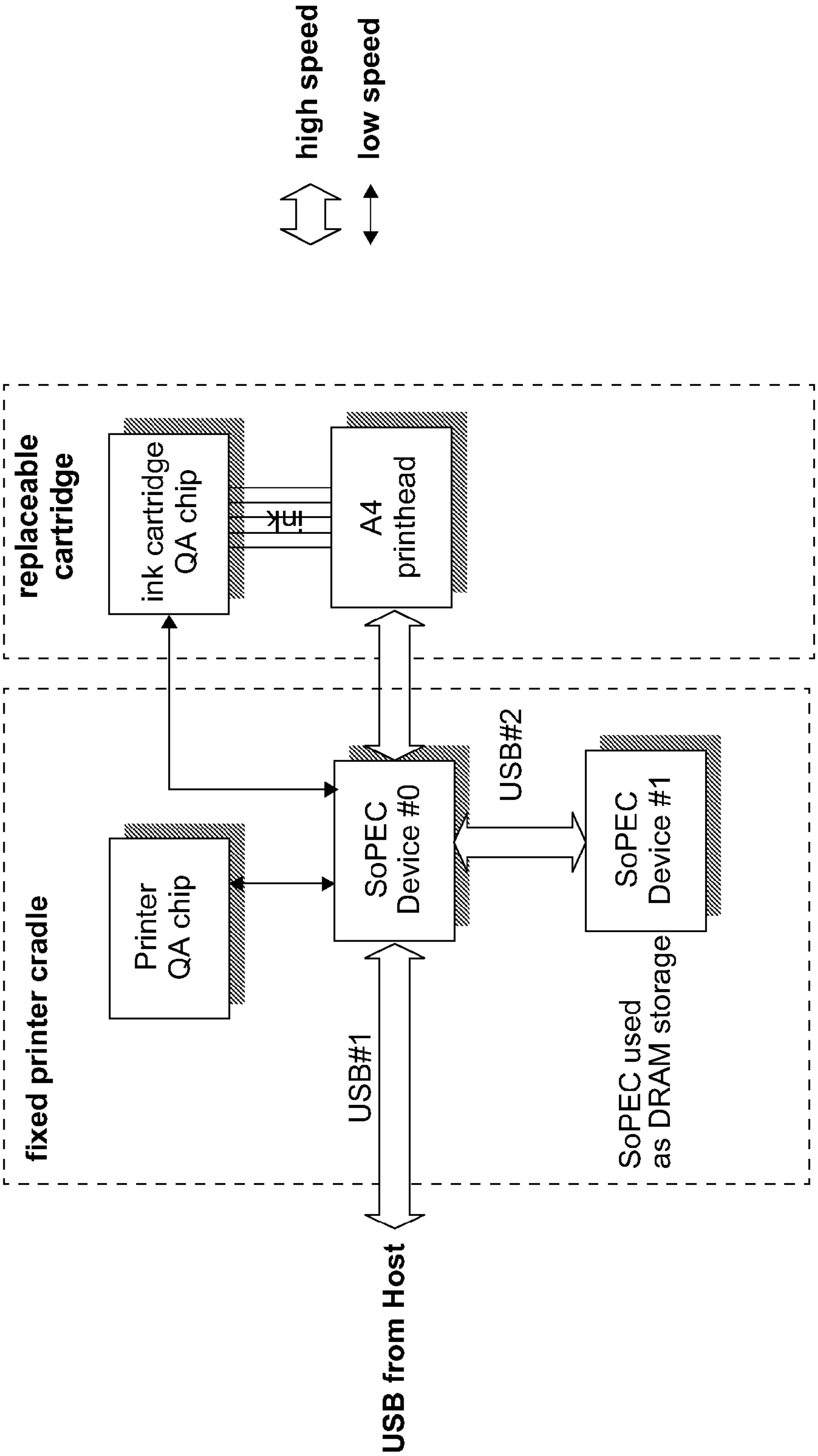
high speed
low speed
replaceable cartridge
ink cartridge QA chip
ink
A4 printhead
fixed printer cradle
Printer QA chip
SoPEC Device #0
USB#2
SoPEC Device #1
USB#1
SoPEC used as DRAM storage
USB from Host

FIG. 6

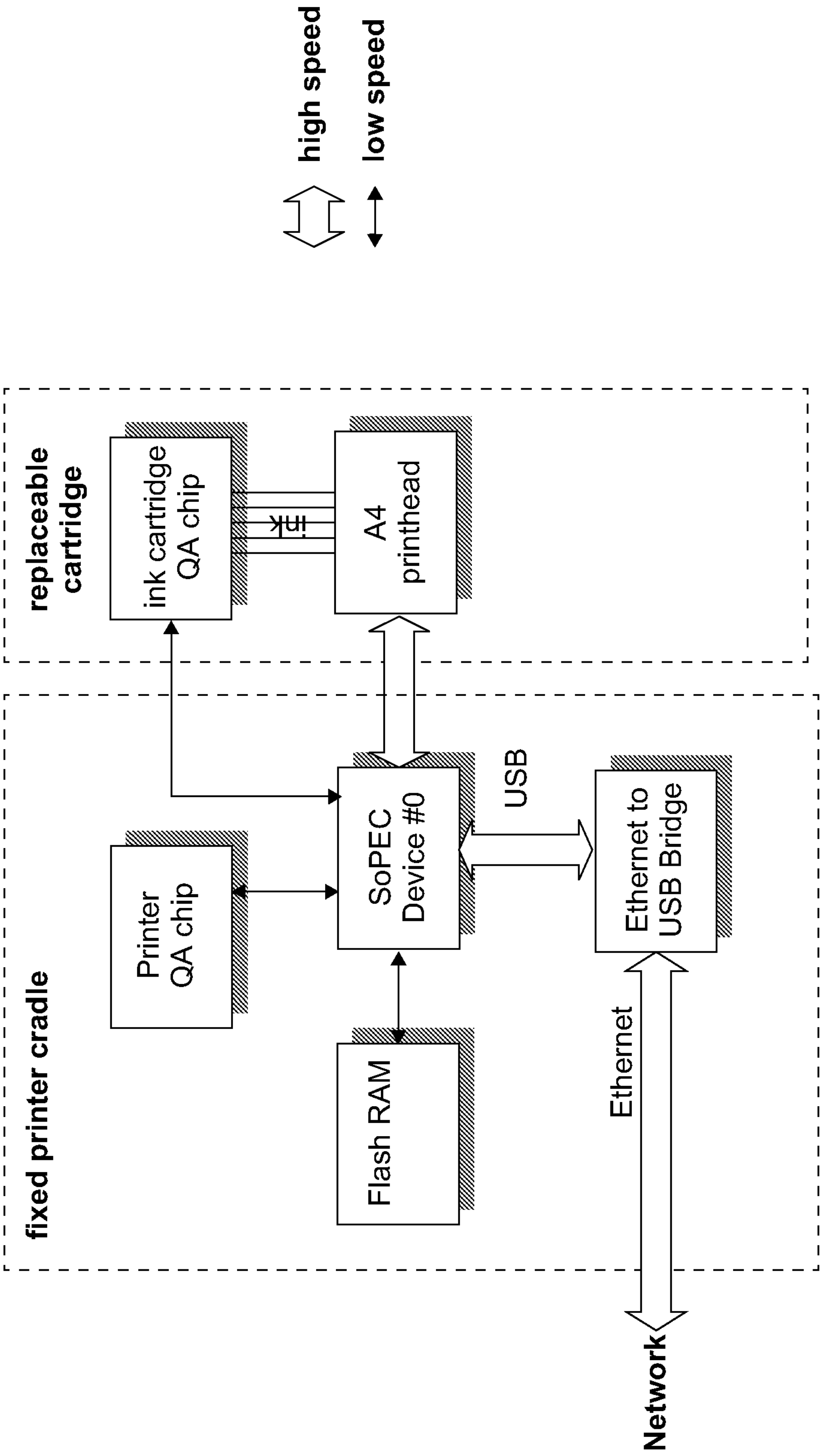
high speed
low speed
replaceable cartridge
ink cartridge QA chip
ink
A4 printhead
fixed printer cradle
Printer QA chip
SoPEC Device #0
Flash RAM
USB
Ethernet to USB Bridge
Ethernet
Network

FIG. 7

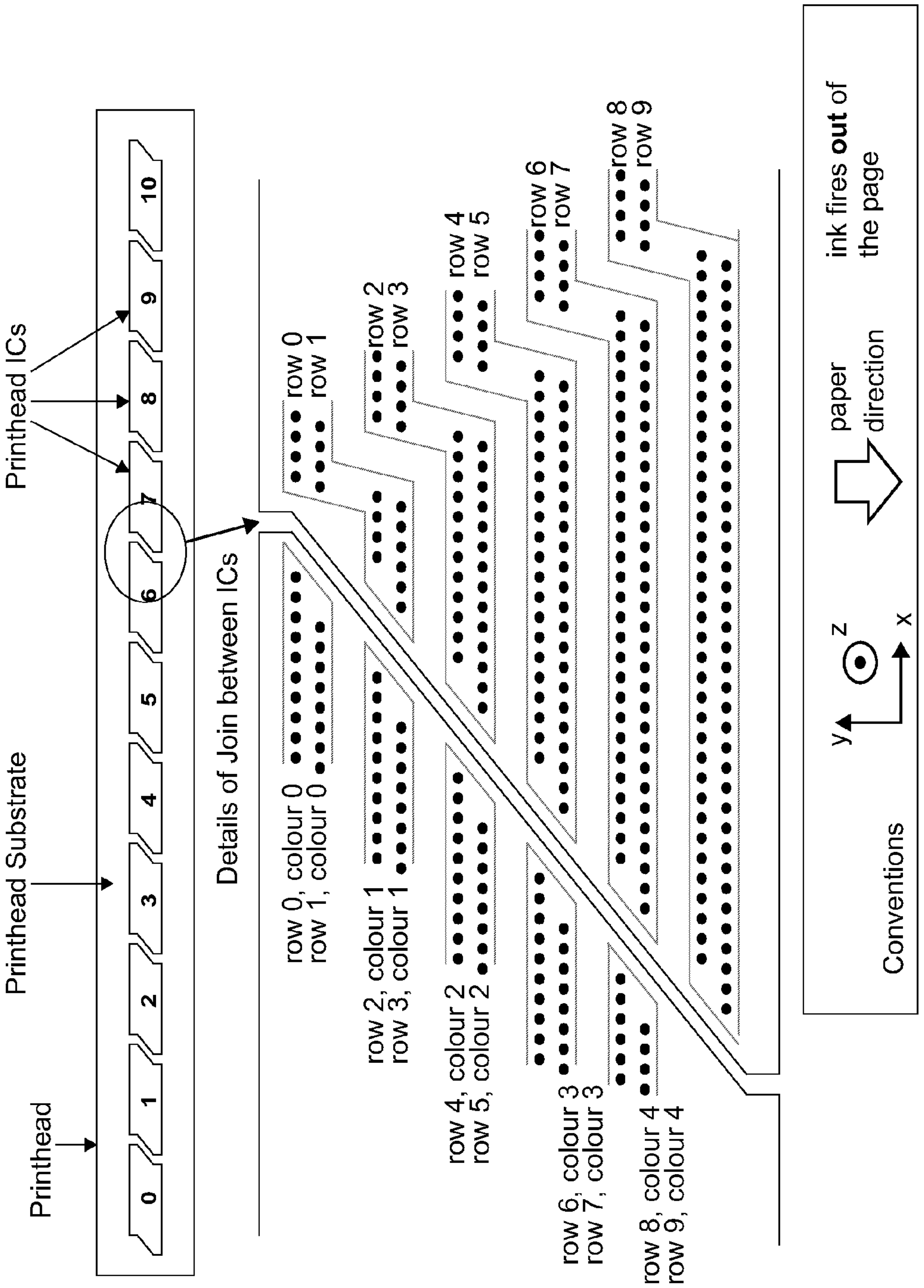
Printhead
Printhead Substrate
Printhead ICs
0
1
2
3
4
5
6
7
8
9
10
Details of Join between ICs
row 0, colour 0
row 1, colour 0
row 2, colour 1
row 3, colour 1
row 4, colour 2
row 5, colour 2
row 6, colour 3
row 7, colour 3
row 8, colour 4
row 9, colour 4
row 0
row 1
row 2
row 3
row 4
row 5
row 6
row 7
row 8
row 9
Conventions
y
z
x
paper direction
ink fires **out** of the page

FIG. 8

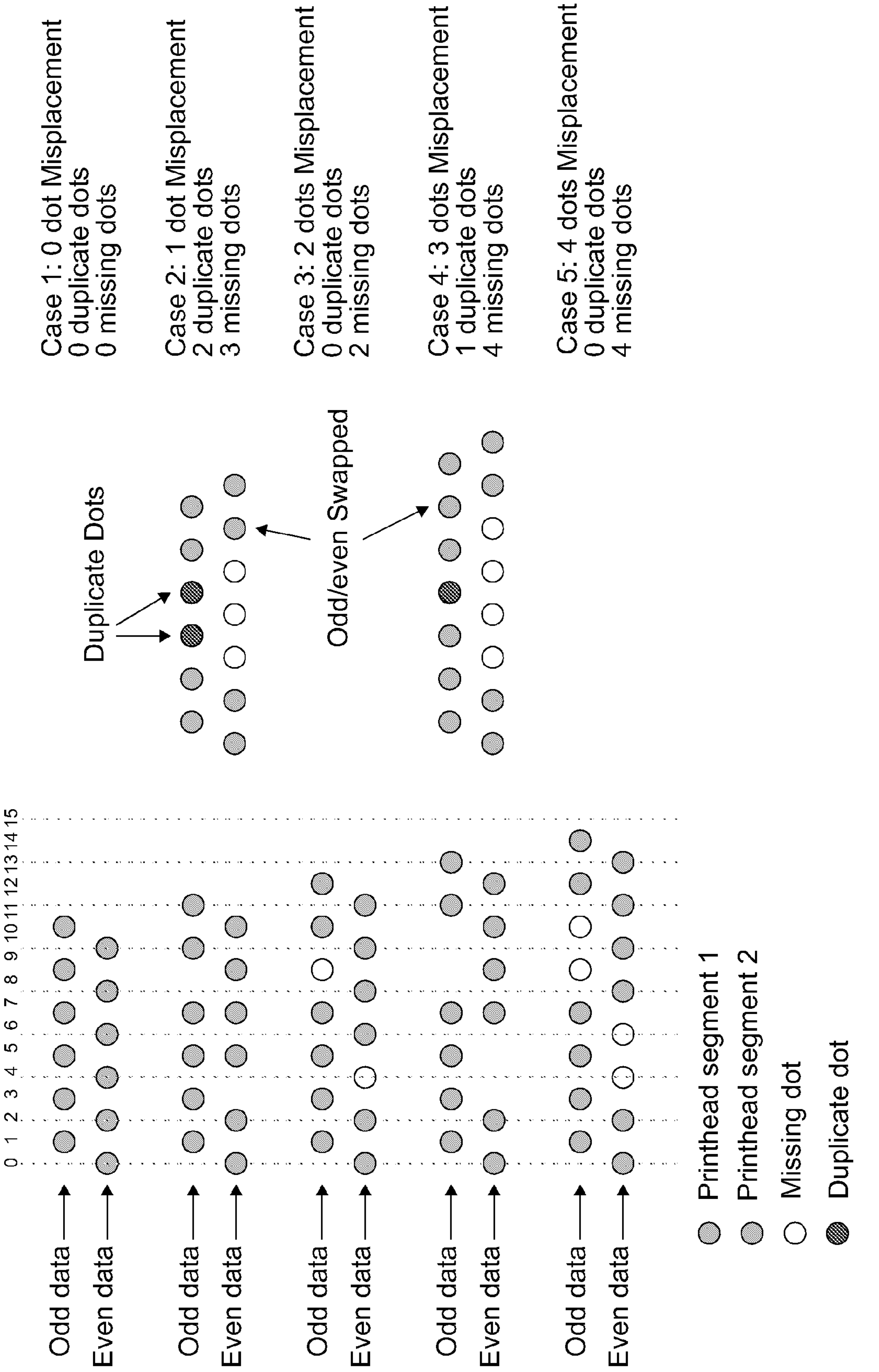
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Odd data
Even data
Odd data
Even data
Odd data
Even data
Odd data
Even data
Odd data
Even data
Duplicate Dots
Odd/even Swapped
Case 1: 0 dot Misplacement
0 duplicate dots
0 missing dots
Case 2: 1 dot Misplacement
2 duplicate dots
3 missing dots
Case 3: 2 dots Misplacement
0 duplicate dots
2 missing dots
Case 4: 3 dots Misplacement
1 duplicate dots
4 missing dots
Case 5: 4 dots Misplacement
0 duplicate dots
4 missing dots
Printhead segment 1
Printhead segment 2
Missing dot
Duplicate dot

FIG. 9

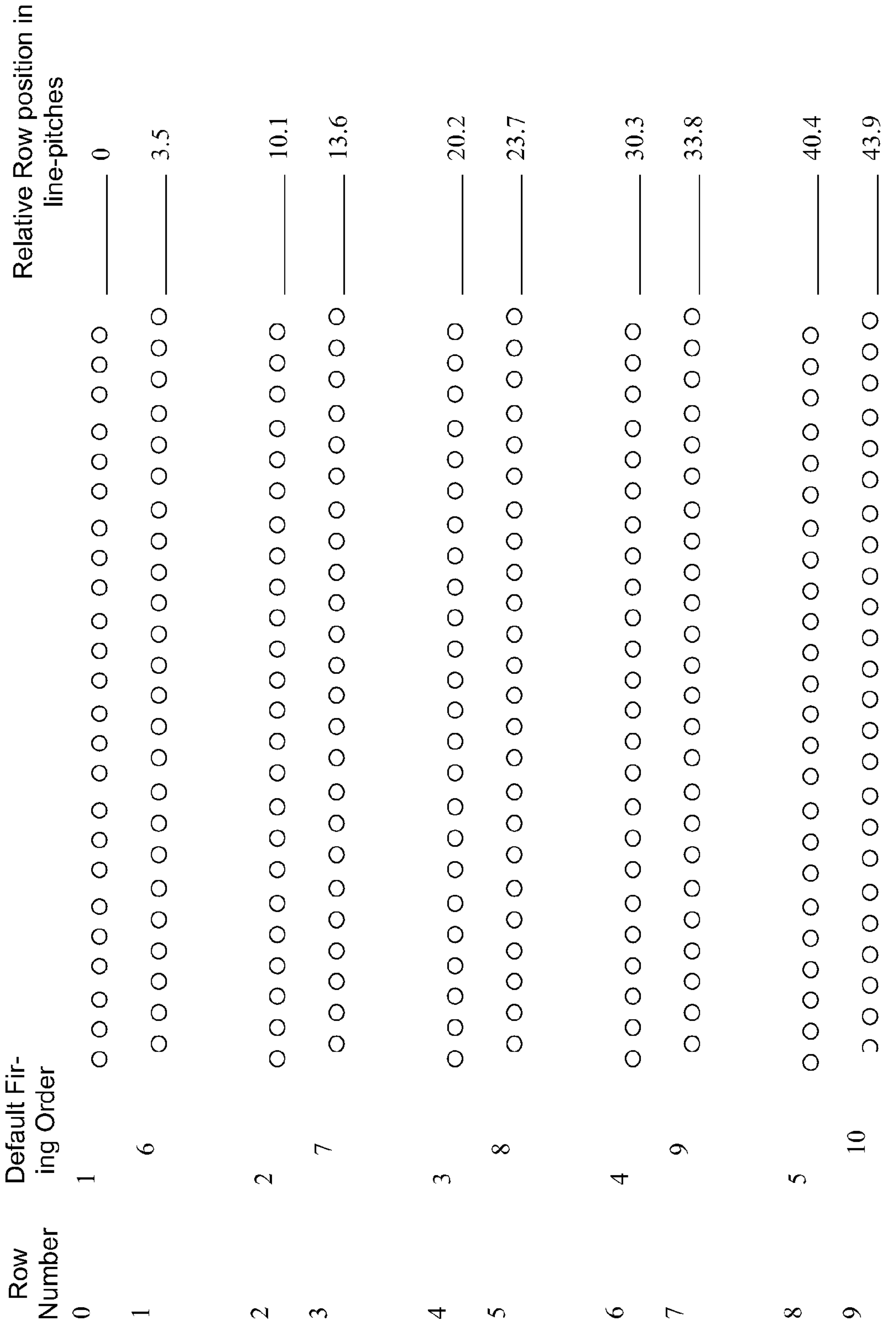
Row Number
Default Firing Order
Relative Row position in line-pitches
0
1
0
1
6
3.5
2
2
10.1
3
7
13.6
4
3
20.2
5
8
23.7
6
4
30.3
7
9
33.8
8
5
40.4
9
10
43.9

FIG. 10

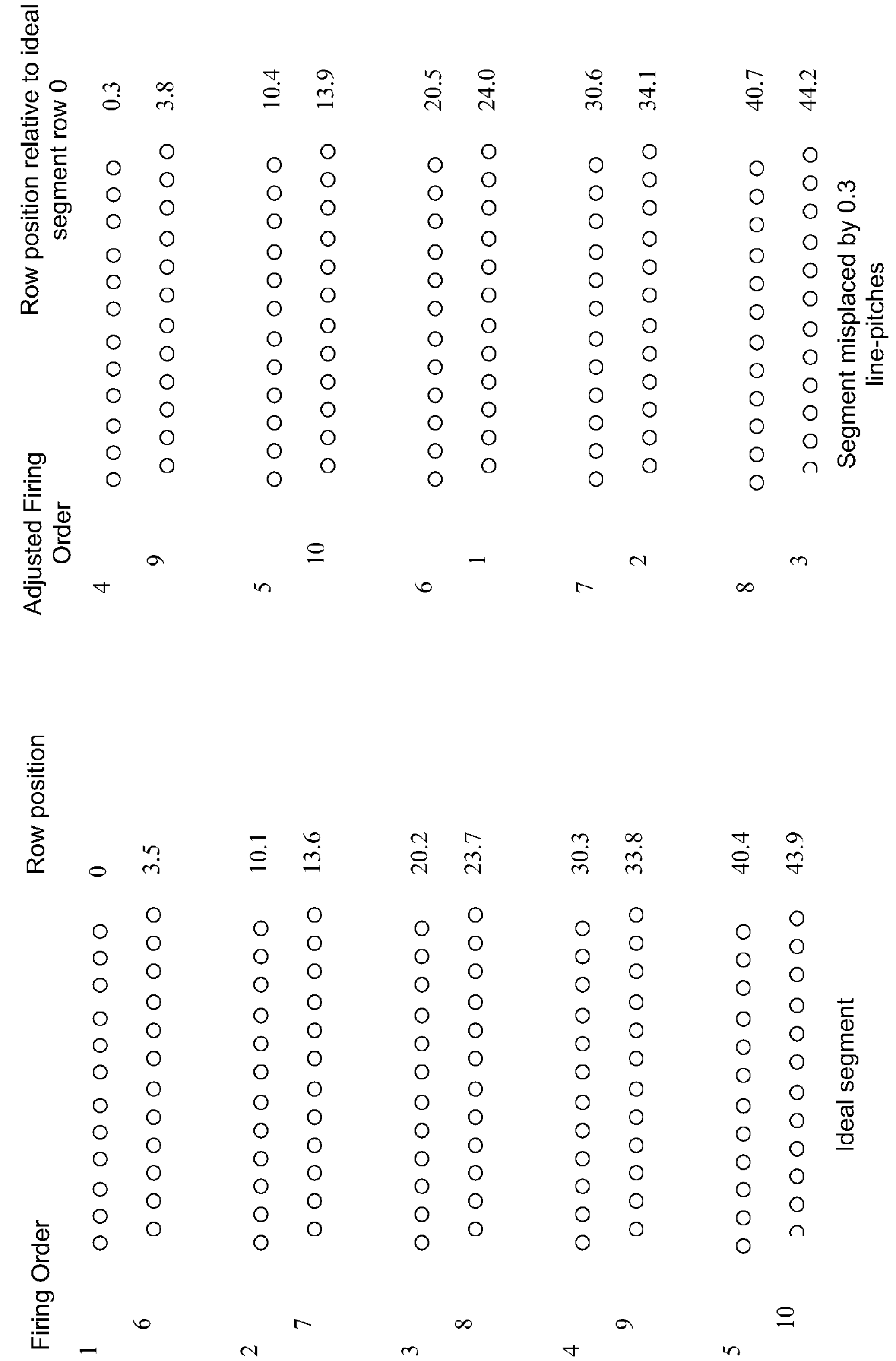
Firing Order
1
6
2
7
3
8
4
9
5
10
Row position
0
3.5
10.1
13.6
20.2
23.7
30.3
33.8
40.4
43.9
Ideal segment
Adjusted Firing Order
4
9
5
10
6
1
7
2
8
3
Row position relative to ideal segment row 0
0.3
3.8
10.4
13.9
20.5
24.0
30.6
34.1
40.7
44.2
Segment misplaced by 0.3 line-pitches

FIG. 11

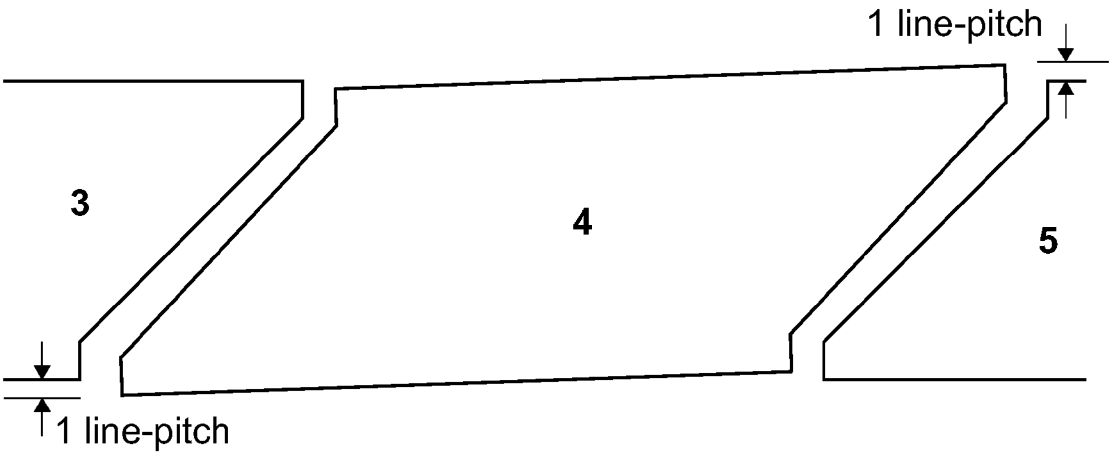
1 line-pitch
3
4
5
1 line-pitch

FIG. 12

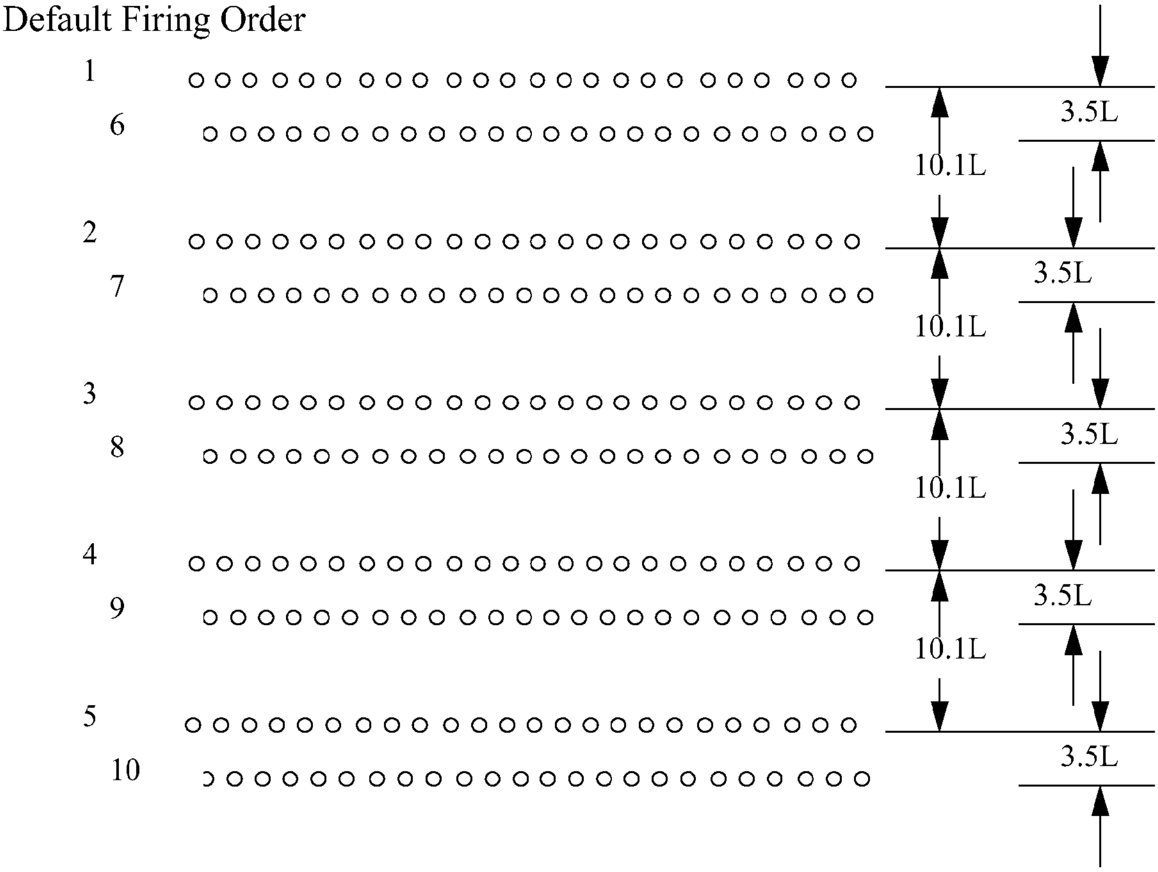
Default Firing Order
1
6
2
7
3
8
4
9
5
10
10.1L
10.1L
10.1L
10.1L
3.5L
3.5L
3.5L
3.5L
3.5L

FIG. 13

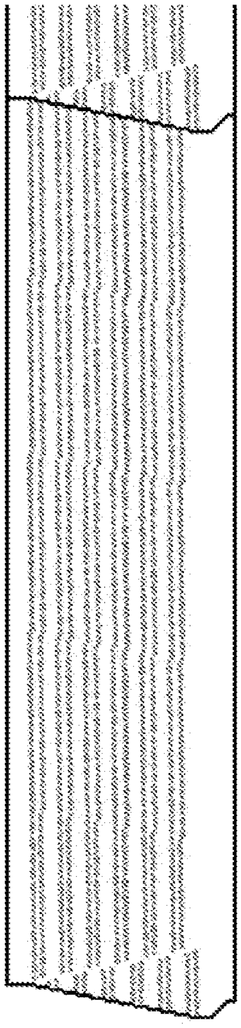

FIG. 14

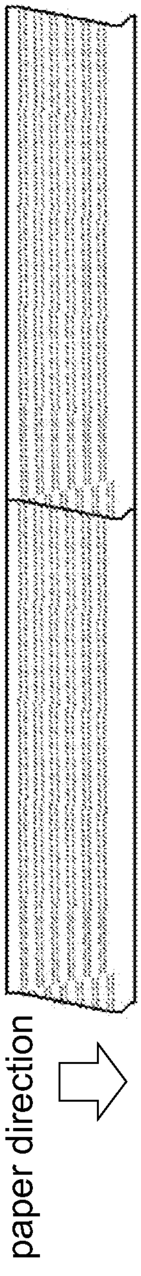

FIG. 15

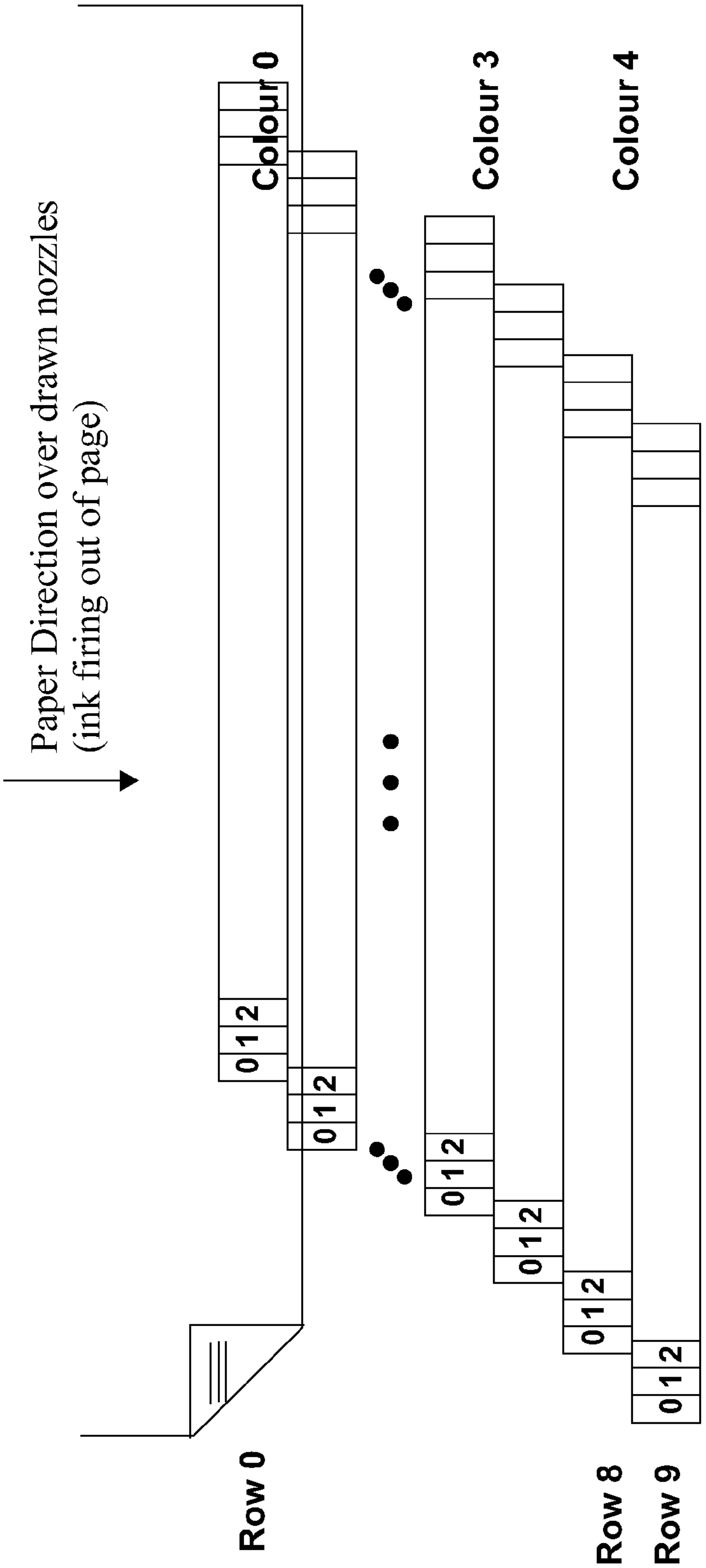
Paper Direction over drawn nozzles
(ink firing out of page)
Colour 0
Colour 3
Colour 4
0 1 2
0 1 2
0 1 2
0 1 2
0 1 2
0 1 2
Row 0
Row 8
Row 9

FIG. 16

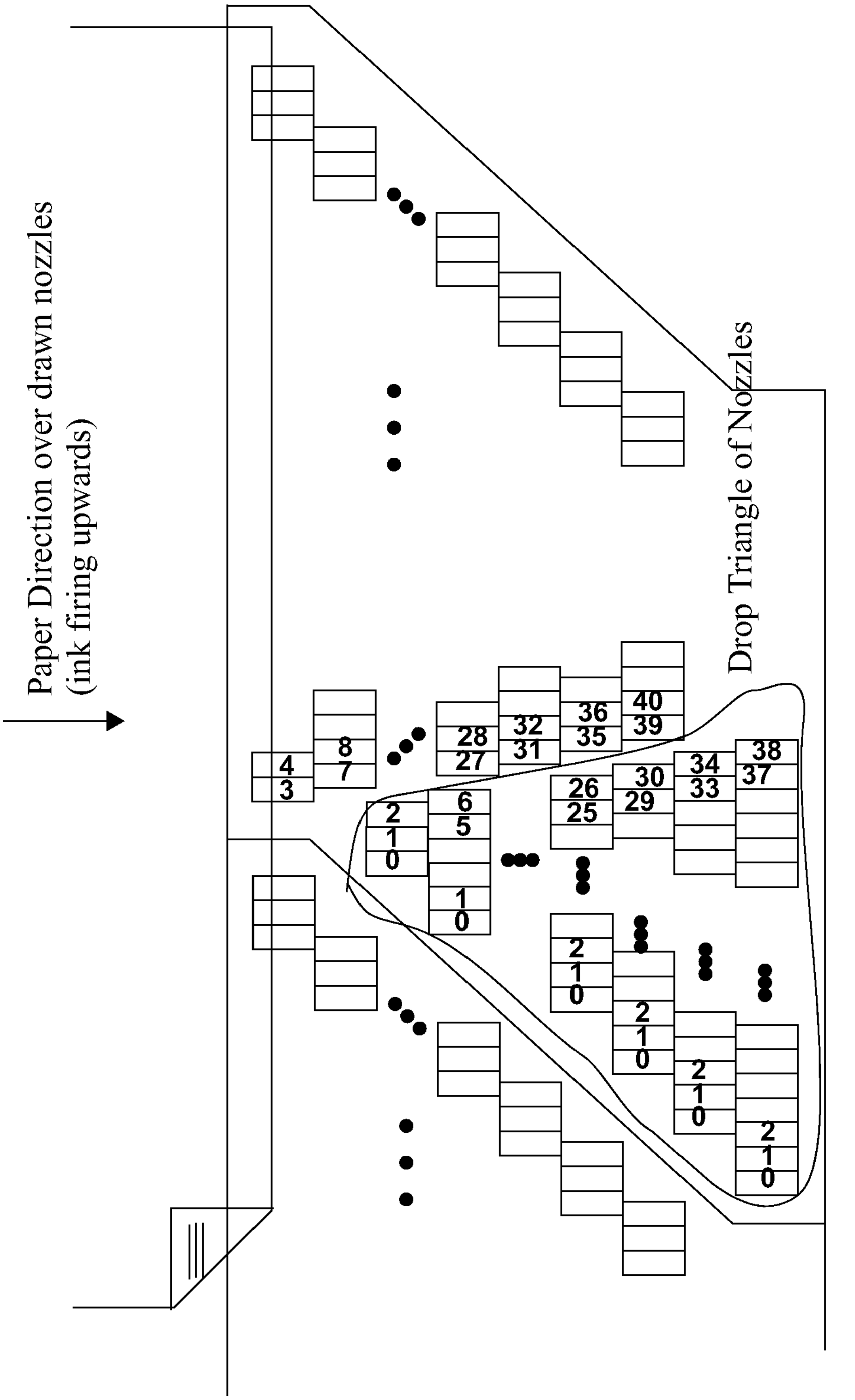
Paper Direction over drawn nozzles
(ink firing upwards)
Drop Triangle of Nozzles
4
3
8
7
28
27
32
31
36
35
40
39
2
1
0
6
5
1
0
26
25
30
29
34
33
38
37
2
1
0
2
1
0
2
1
0
2
1
0

FIG. 17

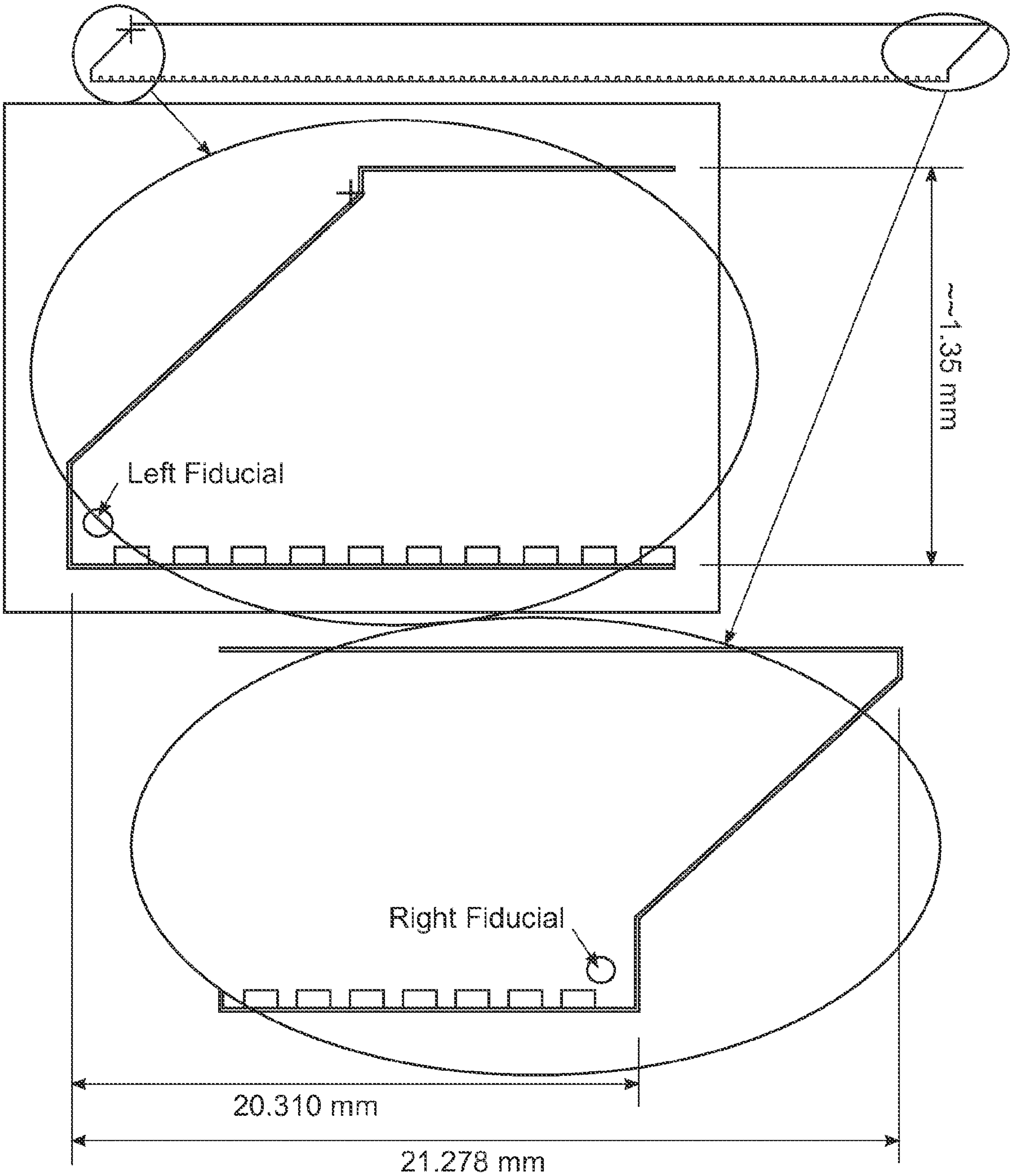

FIG. 18

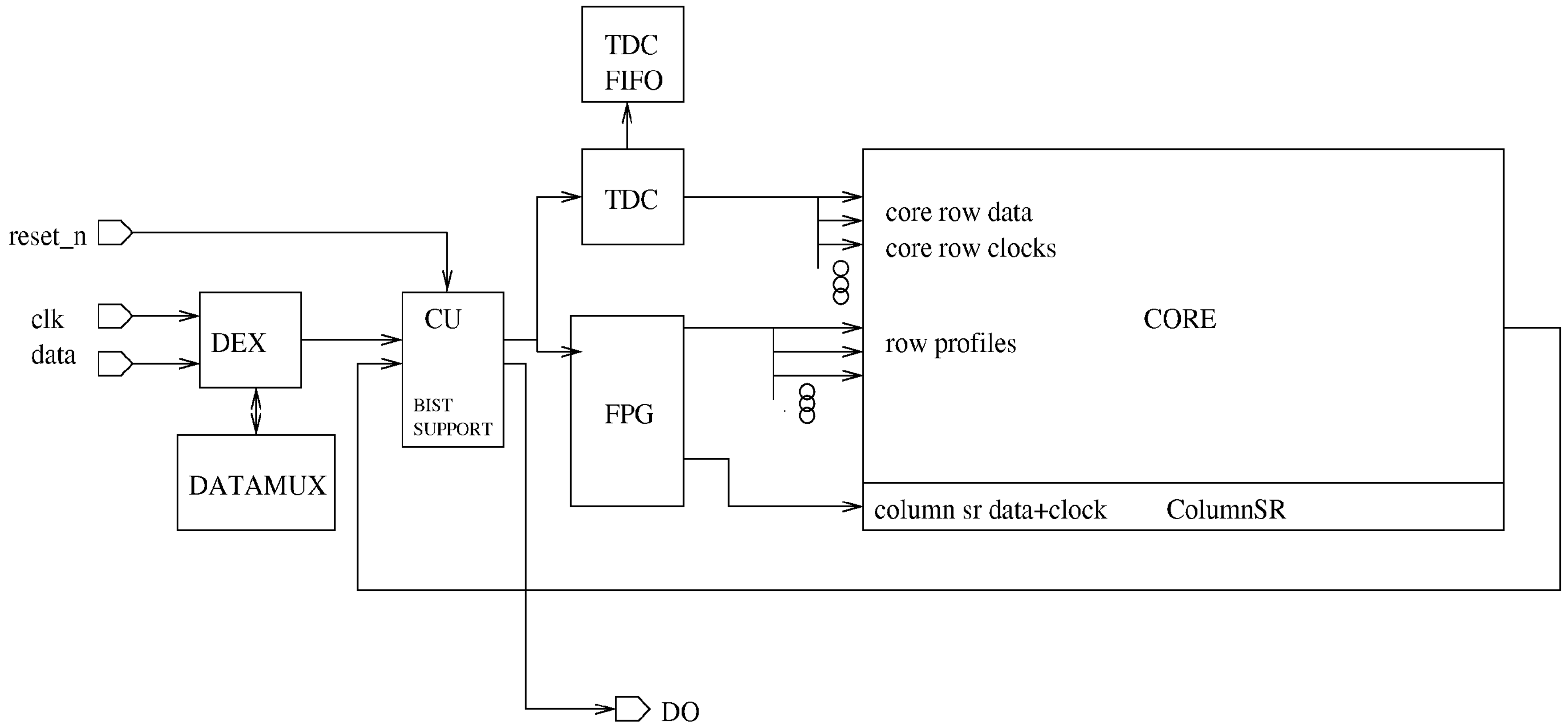
TDC
FIFO
TDC
reset_n
clk
data
DEX
CU
BIST
SUPPORT
DATAMUX
FPG
core row data
core row clocks
row profiles
CORE
column sr data+clock
ColumnSR
DO

FIG. 19

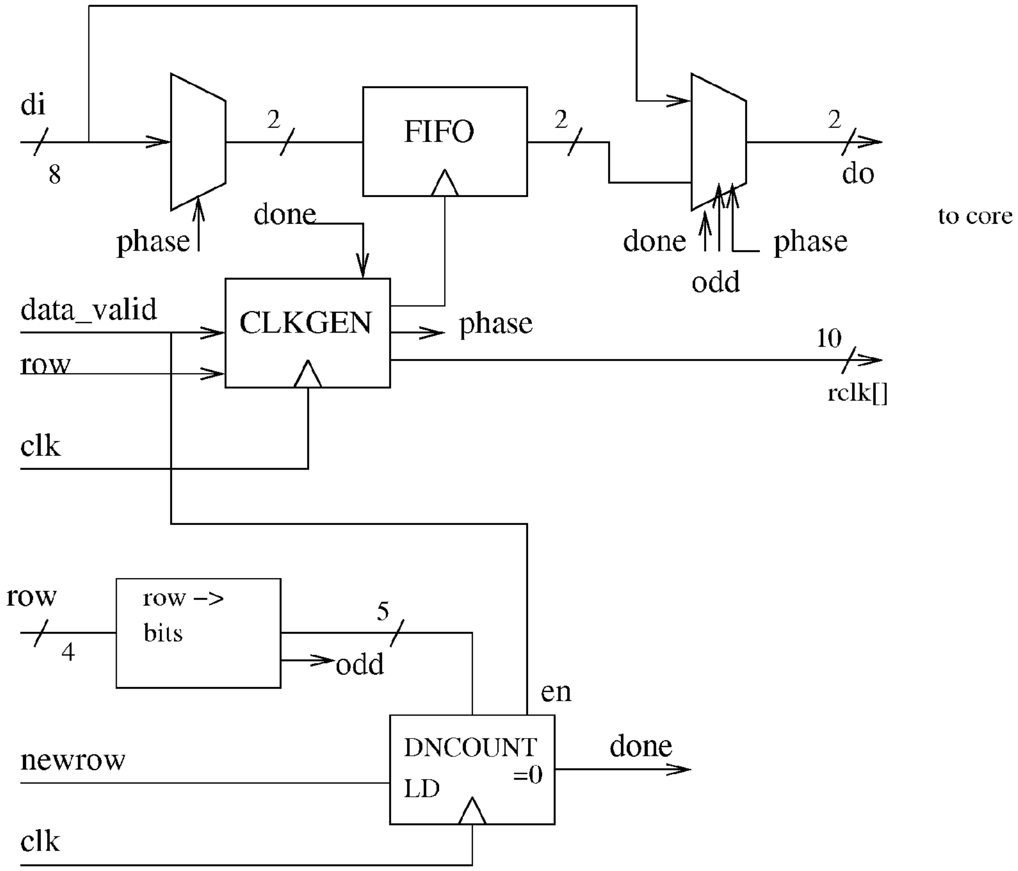
di
8
2
FIFO
2
2
do
to core
phase
done
done
phase
odd
data_valid
CLKGEN
phase
10
row
rclk[]
clk
row
row ->
bits
5
4
odd
en
newrow
DNCOUNT
=0
LD
done
clk

FIG. 20

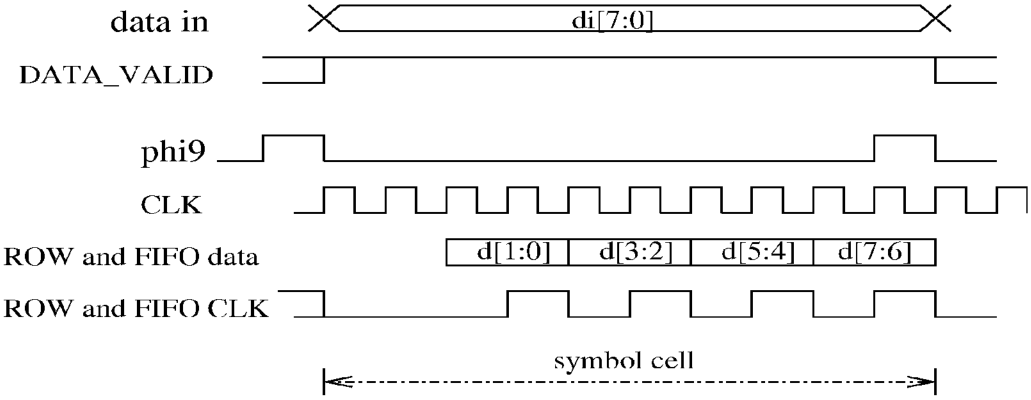
data in
di[7:0]
DATA_VALID
phi9
CLK
ROW and FIFO data
d[1:0]
d[3:2]
d[5:4]
d[7:6]
ROW and FIFO CLK
symbol cell

FIG. 21

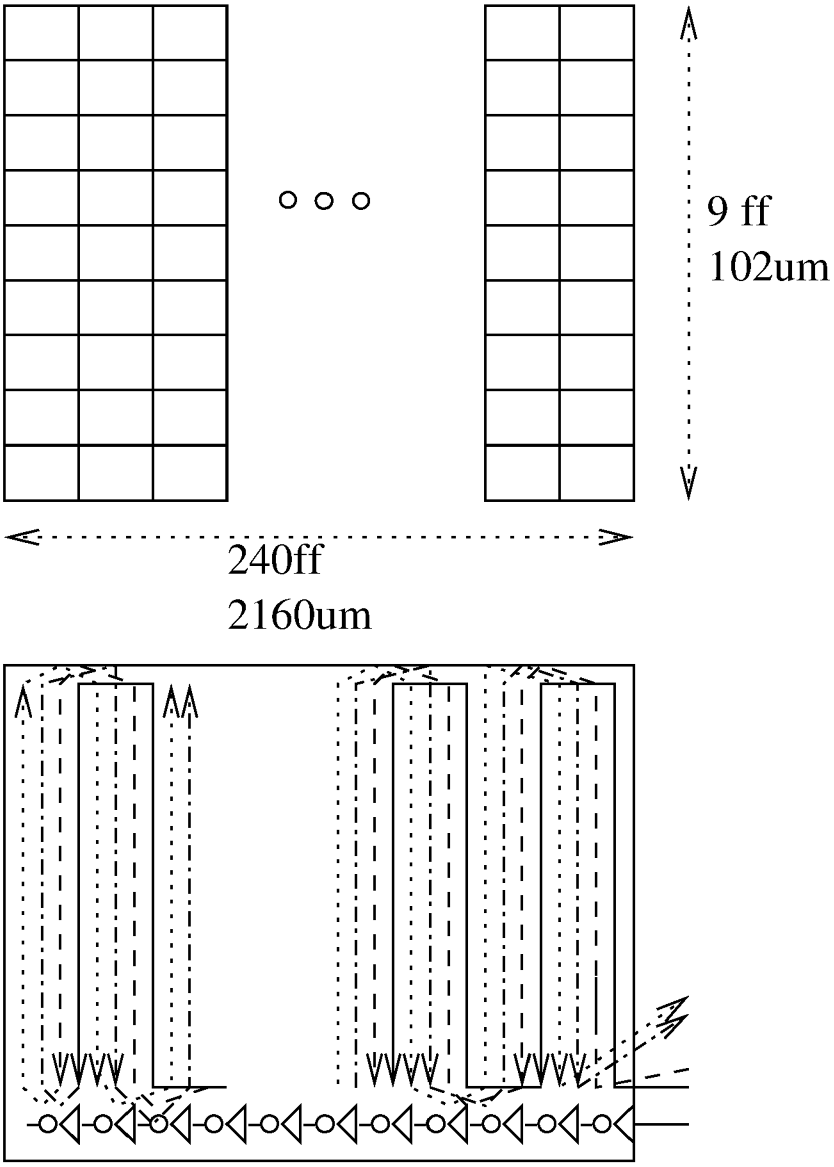

FIG. 22

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 10 19 3974

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X A	EP 1 405 722 A1 (SONY CORP [JP]) 7 April 2004 (2004-04-07) * paragraphs [0034] - [0043]; figures 3-5 * -----	1,2,8-13 3-7	INV. B41J2/07 B41J2/155 B41J2/515
X A	US 2003/122902 A1 (SAKAIDA ATSUO [JP] ET AL) 3 July 2003 (2003-07-03) * paragraphs [0076] - [0095]; figures 4-8 * -----	1-3,8-13 4-7	
X A	US 2002/041299 A1 (LEE SUNG-HEE [KR] ET AL) 11 April 2002 (2002-04-11) * paragraph [0006]; figure 1 * -----	1-3,8-13 4-7	
A	EP 1 375 146 A (OLYMPUS OPTICAL CO [JP] OLYMPUS CORP [JP]) 2 January 2004 (2004-01-02) * paragraph [0009]; figure 1 * -----	1-13	
			TECHNICAL FIELDS SEARCHED (IPC)
			B41J

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Munich	21 February 2011	Kulhanek, Peter

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons
....
& : member of the same patent family, corresponding document

2

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 10 19 3974

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

21-02-2011

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 1405722 A1	07-04-2004	CN 1518502 A	04-08-2004
		WO 03000498 A1	03-01-2003
		JP 4462927 B2	12-05-2010
		US 2004165026 A1	26-08-2004
US 2003122902 A1	03-07-2003	NONE	
US 2002041299 A1	11-04-2002	JP 2002178505 A	26-06-2002
		KR 20020026075 A	06-04-2002
EP 1375146 A	02-01-2004	CN 1491160 A	21-04-2004
		WO 02062581 A1	15-08-2002
		JP 4253504 B2	15-04-2009
		TW 523465 B	11-03-2003
		US 2004041863 A1	04-03-2004

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82