Optical Cross-Connect Based on the Spherical Fourier Cell

David J. Rabb and Betty Lise Anderson, *Senior Member, IEEE*

***Abstract*—A free-space optical cross connect is proposed here that is highly compact and uses digital microelectromechanical systems (MEMS) tip-style micromirror arrays. Based on the Fourier cell, previously described for true time delays, the cross-connect can be strictly non-blocking, or can be of blocking or reconfigurably blocking designs to reduce the MEMS pixel count. Because in the Fourier cell many systems can share a single spherical lens, the hardware count is comparatively low and the system is highly compact.**

***Index Terms*—Fourier optics, optical communication, optical data processing, optical interconnections.**

I. Introduction

TO AVOID the bottleneck of optical-electrical-optical conversion in optical communication systems, it is desirable to have a means to switch a large number of beams among different paths, keeping the signals in the optical domain. Cascading banks of 2×2 optical waveguide switches quickly becomes cumbersome, lossy and hardware-intensive as the number of inputs N increases. The classic free-space optical cross-connect approach uses an analog microelectromechanical systems (MEMS) array of tip/tilt micromirrors combined with an array of input and output fibers [1]–[4]. In these systems, the light from each fiber is directed on a specific micromirror, which is then tilted to the precise angle needed to send it to the appropriate output fiber, usually using a fixed mirror and two MEMS arrays. The MEMS in these cases can tip to any arbitrary angle with in a certain cone.

We have earlier proposed and demonstrated a free-space optical cross-connect [5], [6] based on the White cell. Those devices use two- or three-state digital style MEMS, meaning the micromirrors have two or three stable states, and the angle of tilt is determined by mechanical stops, greatly simplifying the MEMS control electronics and alignment constraints. Further, in the White cell, the accuracy with which a beam is directed to a desired output (for example, to target a single-mode fiber core) is determined by the tilt angles of the White cell (objective) mirrors, which are fixed. The accuracy of the MEMS mirror tilt angles need only be to within a degree or so, since the MEMS only selects which White cells mirror a beam goes to. The White cell-based optical cross-connect is far simpler to align and requires much simpler control.

The White cell is compact but its size does grow as N increases, and the number of MEMS pixels required in the "quartic" version is proportional to N^2 [5] . On the positive side, there is redundancy built in since there are many possible paths to a given output, so that if some micromirrors fail the device can still work.

A binary-style White cell was also described for optical interconnection that uses fewer micromirrors but has no redundancy [5]. This device also requires a number of spot-displacement devices for shifting beams.

In this paper, we propose an optical cross-connect based on the Fourier cell, a different free-space optical system but one that, like the White cell, involves arrays of input beam making multiple bounces through a free-space system, where each beam interacts with a series of MEMS micromirror arrays for independent control. The Fourier cell, however, has the potential to be far more compact, particularly for large numbers of inputs and output, in part because all the beams share a single lens and thus overlap considerably in space. Further, the Fourier cell combines the comparatively low pixel count of the binary White cell with the lack of separate spot shifters of the quartic White cell.

This paper is organized as follows. In Section II, we review briefly the operation of the Fourier cell, and describe how to adapt the Fourier cell to optical cross-connections, driving particular inputs to the required row and column of an array. The problem of combining spots to a particular output spot while avoid fan-in loss is also covered here. Section III is devoted to designs that use fewer MEMS micromirrors; a strictly non-blocking Fourier cell cross-connect is proposed in Section III-A. For a slight increase in MEMS pixels from the blocking design, switch can be reconfigurably non-blocking as described in Section III-B. In Section IV, we summarize our results and give some examples.

II. Operation of the Fourier Cell

A. Fourier Cell

The Fourier cell has been described previously in [7], where it was introduced for the application of providing optical true-time delays for steering phased array antennas. We will briefly review the principle of operation here.

The main component is a spherical lens, Fig. 1. This lens is at the center of a larger sphere (diameter $2f$, where f is the focal length of the sphere). Around half of this outer sphere, on the left, are arranged a series of MEMS micromirror arrays. On the

Manuscript received September 08, 2008; revised January 14, 2009. First published April 24, 2009; current version published July 15, 2009.

D. J. Rabb is with the Electro-Optic Sensors Division, Air Force Research Laboratory, Wright-Patterson Air Force Base, Dayton, OH 45433 USA.

B. L. Anderson is with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus OH 43210 USA (e-mail: anderson@ece.osu.edu).

Digital Object Identifier 10.1109/JLT.2009.2015896

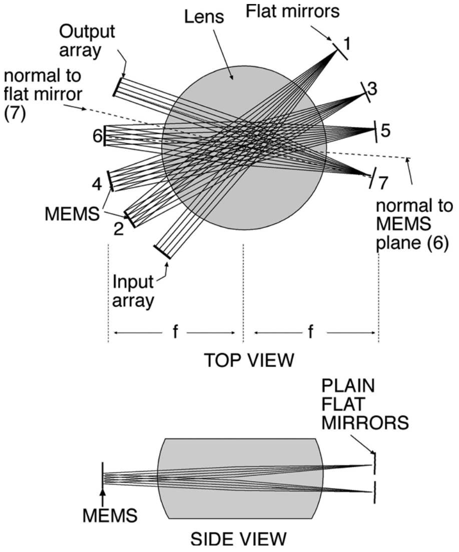

Fig. 1. Basic Fourier cell.

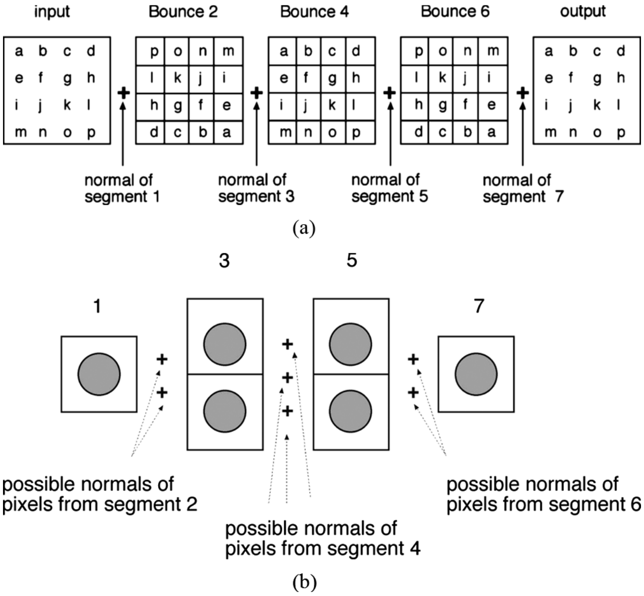

Fig. 2. Spot patterns in the Fourier cell, looking from inside the sphere toward the mirrors. (a) On the even-numbered bounces, the input array is re-imaged with -1 magnification. On the odd bounces (b) the beams are superimposed on the upper or lower flat mirror, depending o the tilt of the pixel from which each beam came.

opposite side of the outer sphere is a series of pairs of mirrors. Four pairs of mirrors are shown. In each pair, one mirror is above the other. An input and output are also shown. The input and output are assumed to each consist of an array of light beams. For example, the array of input beams may come from an array of optical fibers.

It will be recalled that an object placed in the front focal plane of a lens will produce, at the back focal plane, a two-dimensional Fourier transform of the input field [8]. We see in the figure that the spherical lens performs this function on each pass across the lens and again on each return pass. Further, an array of spots at the input is re-imaged, after two transforms, to another array of spots, with magnification -1, for example from one MEMS to the next. Similarly, on the right-hand side, each plain mirror is imaged to the next plane mirror, again with magnification -1.

In operation, the light from the input array of beams, whose axes are all parallel, goes from the input to a single flat mirror at position 1. The Fourier transform process superimposes all the beams in single location on the flat mirror. The mirror is tipped such that the light goes to the MEMS located at position 2, and the lens separates the beam again into individual spots. The MEMS here has pixels that can tip to one of several angles. We choose here three possible MEMS tilt angles, $\pm\theta$ (up or down in the side view) and 0°. Each spot in the light beam array lands on a different pixel. If the pixel for a particular beam is tipped to $+\theta$, that particular beam arrives at the center of the upper mirror in location 3. All beams whose pixels on the MEMS at position 2 were tipped "up" also converge here. For all beams striking pixels tipped downward at position 2, they are all centered on the lower mirror at position 3.

Next we examine the spot patterns and mirror alignments of the original spherical Fourier cell. Fig. 2(a) shows the input, output, and MEMS arrays, viewed as looking at them from the center of the sphere. There are 16 input beams in a 4×4 array in this figure, labeled a-p. The "+" sign between the input and segment 2 represents the intersection of the axis of mirror 1 with the outer sphere. The input array is Fourier-transformed onto mirror 1, and inverse-transformed back to segment 2, where there is a MEMS micromirror array, with 16 pixels. Note the -1 magnification inverts the image.

In Fig. 2(b), we see the flat mirrors on the other side of the sphere, again looking at them from inside the sphere. The first mirror is a single segment. The gray circle indicates the region where all the light beams from the input array are coincident. Between locations 1 and 3 are two "+" signs, one for each of two possible MEMS micromirror tip angles (only two are used here). In location 3, there are two circles. One is where all the beams land that struck pixels tipped up, and the other is for the beams that were sent downward.

Returning to Fig. 2(a), we see a single "+", marking the intersection of the normals of the two mirrors at segment 3 with the outer sphere. These normals are drawn from the centers of the light beams on the upper and lower mirror; therefore, the upper and lower mirrors have slightly different tips.

B. Adapting the Fourier Cell to a Cross-Connect

When the Fourier cell was previously described, for providing true-time delays, the lower mirror in each mirror pair was replaced by some optical device that created a time delay in the beams (compared to if they went to the upper mirror), and those devices acted optically like mirrors. For the optical cross connect, we do not use delay devices, because we want all beams to experience the same latency. We want to deliver each input beam to a particular output location that is programmable. To this end, we realign the lower mirror in each mirror pair in the Fourier plane.

Let us consider the mirror pair in position 3. We now want to produce the functionality that all the beams in the array from the MEMS at location 2 that are sent to the upper mirror are re-imaged onto the MEMS in location 4, just as they were in

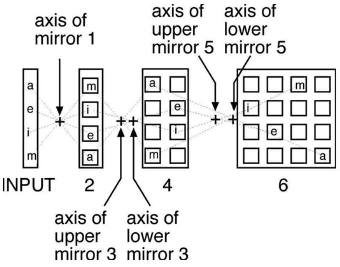

Fig. 3. Beams in a column are individually driven to different columns.

Fig. 2(a). If, however, some beams are sent to the lower mirror, that mirror is tipped slightly to the side, such that when the spots are re-imaged on the MEMS at location 4, those spots form one column over from where they would have formed if they had gone to the upper mirror. This is shown in Fig. 3 for a particular column of the input array. We will drive each spot in this column to a different column.

The operation is as follows. All of the four input beams go to mirror 1, and are re-imaged onto a MEMS device in location 2, where they are still in a single column. At this point, each of the four beams can be sent to either the upper or lower mirror of 3. In this example, beams "a" and "m" are sent to the upper mirror, and their images appear in the first column of the MEMS in location 4. The other two beams are sent to the lower mirror of 3, and their images appear one column to the right. In location 4 each pixel can tip either up, down, or flat, so each of the four beams can appear at either the upper or lower flat mirror in location 5. The axes of these two mirrors are separated by a distance twice that between the axes of the mirrors in location 3, so that for those beams that are to be shifted, they will be shifted two columns to the right instead of one. At this point we have delivered each of the four inputs to a different column. If the process were to continue (say if there were more input beams), the axes of the mirrors in location 7 would be separated by twice as much as the previous odd-numbered location and so on. Thus, N input can be distributed to N different columns using M MEMS chips where

$$N = 2^{(M-1)}. \quad (1)$$

C. Spot Combiner

To make a complete cross-connect, a way must be devised to collapse all the spots in any column to a single spot. We show here how to use the Fourier cell to perform this function.

In Fig. 3, there is one and only one spot in each row. If we could superimpose the rows to a single row, there would be one spot at each location, the desired result. Let us allow the process in Fig. 3 to continue, Fig. 4. We set the MEMS mirror tip angles in the top two rows of location 6 such that they send any beams there to the upper mirror of location 7. The bottom two rows are programmed to send the beams to the lower mirror of 7. The MEMS tip angle needed in each case will depend on whether each beam on 6 is coming from the upper or lower part of 5. The axes of the two mirrors of 7 are as shown. The top rows of 6 are now reproduced in location 8, and the bottom rows are shifted down to rows to overlap them.

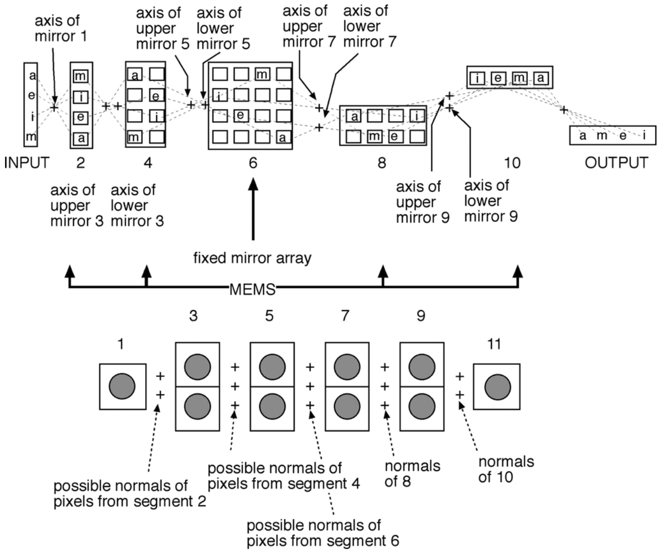

Fig. 4. Spot pattern for switching columns then condensing to a single row.

There will still be only one spot (or no spot) on each MEMS pixel at 8. Each can be tipped to send the beam to the upper or lower mirror 9. These axes intersect with the outer sphere with half the separation of the axes of mirrors 7, resulting in the shifting of one row, putting one on top of the other. At location 10, each light beam in the input column has been sent to a specified spot in a single row. We do not want to put the output directly at 10, however, because the beams will be coming from different angles (upper or lower 9). Therefore, we have one last MEMS that sends each beam to a single mirror at location 11, which then sends them all to the proper output location. At the output, all beams have the same arrival angle, thus enabling efficient coupling in output fibers. Note that if the outputs are detectors, which are not angle sensitive, then steps 10 and 11 can be skipped.

Thus, we have directed four inputs individually and independently to four outputs. The same process can be extended to an arbitrary number of inputs and outputs, so long as the number of inputs is equal to the number of outputs.

The largest MEMS chip that will be required, from Fig. 4, appears to be $N \times N$. The center MEMS (location 6 in this figure) can be static- each pixel will always have the same tip angle since in every case a beam going there came from a known direction and will always be sent to a known fixed direction. Thus, that center device can be a static micromirror array. Assuming that to be the case, the total number of MEMS chips M required for the switch plus combiner is related to the number of inputs/output N by

$$N = 2^{(M)}. \quad (2)$$

D. Comment on Number of MEMS Chips

We observe in Fig. 4 that several MEMS chips of varying sizes are needed. If the MEMS are analog rather than discrete, or at least have a number of possible states, it is possible to combine segments 2, 4. . . , up to but not including the largest MEMS (number 6 in the figure) on a single MEMS chip. The mirrors at the Fourier end of the cell would have their axes intersect the MEMS such that each array of spots (2, 4, etc.) are reproduced on the same chip, shifted by the appropriate number of columns. The MEMS after the "center" one (number 6 in the figure) can similarly be combined. The axes of the various MEMS pixels must still be as shown in the bottom of Fig. 4, requiring therefore many possible MEMS states. This approach uses fewer chips, making it potentially simpler to package, and has the advantage of using all the same chips, rather than an assortment of different sizes. It also has the potential to make the angles of the rays very small with respect to the lens, reducing aberrations and the overall size of the apparatus, at least at the MEMS side of the sphere (the spreading out of the Fourier mirrors may increase the angles too much in certain cases). The drawback is increased complexity of the MEMS drive circuitry.

The cross-connect described so far is strictly non-blocking, but requires large MEMS micromirror arrays. In the next section, we show how to reduce size of the MEMS arrays significantly by incorporating half-row shifts, at which point, however, the switch becomes blocking. A third configuration, using grouping, restores the property of being non-blocking, in this case reconfigurably non-blocking, while still keeping the required MEMS array size small.

III. Reducing the Size of the Required MEMS

A. *Using Half-Row Shifts*

So far, in the method of creating a cross-connect based on the Fourier cell discussed in Part II, the size of the largest MEMS required is $(p)N \times N$, where N is the number of inputs and outputs, and p is the pixel pitch. For a large cross-connect, this may be prohibitively large. In this section, we discuss an improvement to the optical switch that reduces the size of the MEMS to $1.5\ pN^{3/4}$.

The top half of Fig. 5 shows the MEMS segments again as viewed from the center of the system. For this case the MEMS devices are three-state (pixels can be tilted to $-\theta$, flat, or $+\theta$). The segments (MEMS and flat mirrors) are each flat and are on a continuous spherical surface with the normals of each going through the center of the ball lens as before. The figure shows the case where there are sixteen inputs, arranged in a 4×4 square. We will follow the paths of four specific inputs, indicated by gray circles, to illustrate the operation of a cross-connect with "grouping."

Also shown in the top of the figure are the locations of the normals of the Fourier mirrors. Rays coming out perpendicular to Fourier mirrors come to a point on the MEMS side. The same is true for the lines perpendicular to MEMS pixels: they come to a point on the Fourier mirror side. The Fourier mirrors are shown in the lower half of the figure as viewed from the ball lens. The letters inside the large spots correspond to those beams that visit that part of the Fourier mirrors. Which part of the Fourier mirror (top or bottom) is visited at Bounce 3 (determined by the tip of the various pixels at 2) determines the point about which a particular beam is imaged in going from one MEMS chip to the next.

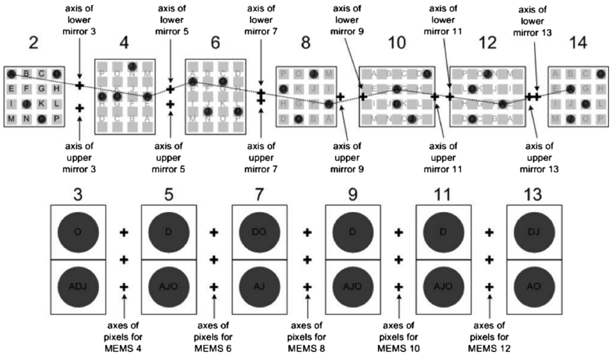

Fig. 5. Using half-row shifts, either up or down (first bounces) or left or right, (second half of bounces) the size of the MEMS chips needed can be reduced. The gray letters indicate where the pixels would be imaged without the half-row shifts. The black circles follow four inputs, A, D, J, and O, to show how they can be independently routed to arbitrary outputs.

In this example, the MEMS at Bounce 2 is 4×4, the same dimension as the input array. The MEMS at Bounce 4, however, is 5×4. Since the progression is from an even to an odd number of rows every spot from Bounce 2 must be shifted either up or down a little to land on a pixel in Bounce 4. A given beam can be moved to one of two different rows when imaged again at segment 4, depending on whether it struck the upper or lower mirror on Bounce 3. Shown faded on each of the MEMS segments are the previous default image locations. That is, these would have been the beam locations throughout the system if the Fourier mirrors were not tilted. In this design, a beam is shifted either up or down one and a half rows relative to the default image (shown faded), where a half row lies between rows of the default image. Either shift puts a beam on a new row of pixels at Bounce 4. The Fourier mirrors at 5 are tipped such that a given row is shifted up or down one row at MEMS 6, and the Fourier mirrors at 7 are tipped to shift each spot up or down a half row, again relative to the default image. The mirrors at 9, 11, and 13 are used to shift the columns by ± 1.5, ± 1 and $\pm\ 0.5$ pixels respectively relative to the default image. The total shift of a beam row can then be any integer from -3 to 3 and its column can be shifted similarly.

For each array we will define the default position P as being position (4, 4) (the position in the array of input P if it is never shifted) and default A as being (1, 1). Let us follow Beam A, which in this example is to be directed to the output position (2, 2). Thus, its position must be shifted down one row and one column to the right at the output. The row of the beam starts at 1. The pixel at bounce 2 is tilted to send Beam A to the lower Fourier mirror at Bounce 3, which causes a shift of 1.5 rows such that the beam goes to the center row at segment 4. Note that the image of MEMS 2 is inverted, so while the beam is moving "up" on Bounce 4, it is really down a row and a half with respect to the input and output. Thus we call this a shift of -1.5 rows. Because the rows of the pixels at Bounce 4 are between the images of the rows at Bounce 2, Beam A strikes a pixel at Bounce 4. This pixel is tilted downward, sending Beam A to the lower Fourier mirror on Bounce 5. This then shifts Beam A to row 1.5 (a$+1$ shift with respect to the input and output arrays) at segment 6. The MEMS at 6 sends Beam A to the lower Fourier Mirror at Bounce 7, which provides an additional shift of -0.5 rows. Note that the

MEMS at Bounce 8 is now 4 × 4 again. Thus, the half-row shift puts the beams on a pixel in row 2 at bounce 8. The total row shift is $-1.5 + 1 - 0.5 = -1$ row as required. Input A started in row 1 and ended up in row 2, the intended destination row. Now, however, the column must be shifted to send Beam A to the desired output point.

The column is shifted in a similar fashion. Now the MEMS in Bounces 10 and 12 are 4 × 5 (instead of 5 × 4 as in Bounces 4 and 6). Beam A is sent to the appropriate Fourier mirrors at Bounces 9, 11, and 13, such that the beam is first shifted by $+1.5$ columns (remember the inverted image), then -1 columns, and finally $+0.5$ columns. The MEMS at Bounce 14 is 4 × 4 again, and Beam A, with total column shift of $+1.5 - 1 + 0.5 = +1$ lands on a pixel in column 2 as desired.

Following the case of the input beam as position "O," it begins at (4, 3) and is to end up at (1, 4). Thus, its row is shifted -1.5 rows, -1 row, and -0.5 rows for a total row shift of -3 rows. Similarly, the column is shifted by $+1.5$, -1, and $+0.5$ for a total shift of $+1$ to column 4.

There is at least one way to get any of the possible net shifts, with two ways to get a shift of zero ($+1.5-1-0.5$ or $-1.5+1+0.5$). The Fourier mirror visited on each pass is chosen so that the resultant shift puts the beam closer to its desired final position than the other possibility. In the case of a zero total shift the mirror for the first shift will be chosen to keep the beam closer to the center of the array so as to limit the size of the largest array.

The arrays shown are no larger than 5 × 4 or 4 × 5; this comes about since beams on the outside rows or columns of the array are automatically shifted inward. For example, a beam in the top row has a net shift between 0 and 3, so for the first shift it can automatically get the $+1.5$ shift, eliminating the need for a row of pixels at -0.5. Similarly beams in the bottom row get the -1.5 shift and no row is needed at 5.5, so there are five rows needed at 0.5, 1.5, 2.5, 3.5, and 4.5. The last shift is ±0.5 so the beam must be within a half row of its output position so there are the same five rows for the next MEMS array. Since the columns are shifted in the same manner the columns occupied will be the same as for the rows.

Following the beam incident at A, we see in Fig. 5 that at each shift the beam needs to visit the lower half of the corresponding Fourier mirror. Looking at the beam locations of A at the Fourier segment we can see that each of the MEMS pixels at bounces 4, 6, 8, 10, and 12 where the beam is incident will have a tip of $-\theta$.

In general for an N-input system configured in an $N^{1/2} \times N^{1/2}$ array, in order to shift from any input row, R_{in}, to any output row, R_{out}, the magnitude of the shifts required are given by

$$R_{\text{out}} = R_{\text{in}} \pm \frac{(N^{1/2}-1)}{2} \pm \frac{N^{1/2}}{4} \pm \frac{N^{1/2}}{8} \cdots \pm 1 \pm 0.5. \quad (3)$$

Similarly, the column shifts are given by

$$C_{\text{out}} = C_{\text{in}} \pm \frac{(N^{1/2}-1)}{2} \pm \frac{N^{1/2}}{4} \pm \frac{N^{1/2}}{8} \cdots \pm 1 \pm 0.5. \quad (4)$$

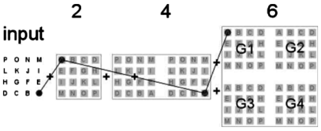

Fig. 6. Moving beams into separate groups.

The net shift of any beam will be between plus and minus $(N^{1/2} - 1)$. The shifts count down in a binary sequence (each shift is half that of the next) ending with the smallest shift of ±0.5. The difference between the smallest shifts possible sets the resolution, $0.5 - (-0.5) = 1$, which is the resolution required. The largest shifts are slightly smaller than twice that of the second largest, so that a net shift of zero is possible.

Since the shifts count in a binary sequence the total number of shifts required is determined using a log in base 2. For the rows there are $2N^{1/2} - 1$ possible shifts required between plus and minus $(N^{1/2} - 1)$, requiring $\log_2(2N^{1/2})$ shifts. Since the columns are shifted similarly the total number of shifts required S to get a beam to any of the desired locations can be determined using

$$S = \log_2(2N^{1/2}) + \log_2(2N^{1/2}) = \log_2 N + 2 \quad (5)$$

or $S = 6$ *shifts for an* $N = 16$ input system.

In the case shown, the largest MEMS dimension is 5 (pixels). For a larger number of inputs the shifts can be ordered so that the largest dimension of an array A is given by

$$A = p\left(N^{1/2} + \frac{N^{1/2}}{4} + \frac{N^{1/2}}{4}\right) = 1.5\, pN^{1/2} \quad (6)$$

where p is the pixel pitch. Having the second largest shift occur first and the largest second does this. Since the second largest shift is of size $N^{1/2}/4$ a beam can be no more than that many rows or columns outside the input array size. After the second shift, the remaining shifts can add up to no more than $N^{1/2}/4 - 1$, so in this array and in each subsequent the row or column is no more than that amount outside the input array size. So the largest array will occur after the first shift, where the row/column could be $N^{1/2}/4$ outside on either side.

In the example of Fig. 5, we chose to follow four input beams whose output locations were chosen such that the beams will never be incident on the same pixel. This is because no two beams have the same input column or output row. In the worst case we could have four beams in one input column going to the same output row, all the beams would overlap after their rows were adjusted to match the output. Thus this design is not non-blocking. We can correct this by sending the beams to different areas or groups to adjust their positions. We will demonstrate this next.

B. Using Grouping

A non-blocking version of the optical switch, albeit it one that requires larger MEMS arrays, can be implemented by grouping the beams in the manner described next. There are four different groups for a 16-input array. Each group is an image of the initial input array, and the groups are separated so that they are in separate areas and their pixels do not run into one another. The number of groups, G, required for an N input system is

$$G = N^{1/2}. \quad (7)$$

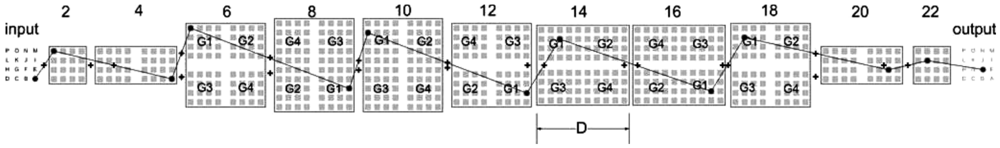

Fig. 7. MEMS arrays for a 16–input switch.

Fig. 6 shows how a beam in the input array could be moved to one of four different groups. First the input array is imaged onto a MEMS (segment 2) whose pixels can be tilted to send an incident beam to either the top or bottom of the next Fourier segment. The top and bottom Fourier mirrors have tips that create two different images next to each other that do not overlap (segment 4). The next Fourier segments create two different images of the MEMS array pair from segment 4; these two images again do not overlap and are above one another at segment 6. At segment 6, the four groups are labeled G1, G2, etc., each an image of the input array.

In the figure, the path of a beam at A is followed as it is moved into group G1. The number of MEMS bounces required to duplicate the inputs into the $N^{1/2}$ groups needed is $\log_2 N^{1/2}$. Now we have reduced the problem to that of Fig. 5. Beams are sent to the appropriate groups, such that within a group no two beams have the same input column or output row.

In Fig. 7, the MEMS arrays for a complete switch are shown; here, the separate groups are labeled and the bounce pattern for a beam is shown. Again a beam at input A location is followed as it is moved into group G1; next the location within the group is adjusted to match that of the desired output location, and finally the groups are recombined to form one output array. The number of bounces required to combine the groups is the same as to separate them $\log_2 N^{1/2}$. The total number of MEMS required M for the switch are those required for the shifts S one for input/output, the separating and then recombining of groups is

$$\begin{aligned} M &= S + 1 + \log_2 N^{1/2} + \log_2 N^{1/2} \\ &= 2\log_2 N + 3. \end{aligned} \tag{8}$$

The largest MEMS dimension for any one segment in the case shown in Fig. 6 is ten pixels. The equation for the largest MEMS dimension A for a larger number of inputs N and pixel pitch p is given by

$$A = 1.5\,pN^{3/4}. \tag{9}$$

Comparing to (6) for the blocking switch, the largest MEMS is now only slightly larger but now the switch is (reconfigurably) non-blocking.

These larger MEMS have many more pixels than beams present. For the case shown in Fig. 4, we know that for the first half of these bounces after groups are separated that the row is being adjusted while the column remains constant. By the way the groups were determined it is also known that no two beams in the same group occupy the same input column. This means that we could drive the entire column of pixels with one signal. Similarly, we know that no two beams occupy the same output row and that for the second half of bounces the row remains constant, so one signal could be used to drive the row. So for any one MEMS only N drive signals are required. This is important for reducing the required pin-out for the MEMS packaging.

IV. Discussion

We have presented several configurations of an optical cross-connect based on the Fourier cell. The advantages of this approach are as follows. First, the MEMS are very simple, since they are digital, with two or three discrete states. Thus they are simpler to manufacture and particularly to control compared to analog MEMS. Alternatively analog MEMS can be used, reducing the overall MEMS chip count. Second, because all the beam paths share a single lens, the apparatus can be very compact. Third, many cross-connects can be cascaded around the same lens, similar to the method of cascading time delay devices in [7]. Third, the cross-connect will be comparatively easy to manufacture. Fourth, the method of combining beams here will avoid fan-in losses. This is a direct result of the requirement that only one input can be sent to a particular output location. Regardless of the original location, each beam arrives at the appropriate output with the same angle and beam size.

In the simplest version, the size of the largest MEMS chip was N^2 pixels for N inputs and outputs. That design was strictly non-blocking but required large MEMS micromirror arrays. We then showed how to reduce this to $1.5(N^{1/2})$, a large decrease in MEMS size but at the cost of producing a system that is no longer non-blocking. One can increase the size of the MEMS to $1.5(N^{3/4})$ and produce a reconfigurably non-blocking cross-connect. If strict non-blocking is required, then the original design is needed. For a switch with 256 inputs and outputs, the largest MEMS is 256 × 256, or 65 536 pixels, whereas the reconfigurably non-blocking system requires a maximum MEMS size of only 96 × 96, or 9216 pixels, with the added simplicity that in each of those MEMS, the wiring is greatly reduced because either the rows or columns can be wired together. The MEMS are greatly simplified, and it should be noted that all the MEMS in a given bounce (e.g., all four MEMS in bounce 6 of Fig. 7) can be on a single substrate. The tradeoff for smaller MEMS is the requirement of more of them, and the addition of more bounces.

Continuing a 256 input/output system design, the largest MEMS size for a 150-μm pixel pitch would be 14.4 × 14.4 mm. The system would require a total of 19 MEMS segments in addition to input, output, and 20 Fourier segments. The total number of segments is then 41, if we allow two systems (total 82 segments) to use the same spherical lens, each segment would subtend 4.39°. This along with 14.4 mm MEMS size gives us a total system diameter of 189 mm. Such a system would require MEMS tilt angle accuracies $\sim$ 1 mrad, and Fourier segment tilt accuracies of ~ 10 μrad.

References

[1] A. Olkhovets, P. Phahaphat, C. Nuzman, D. J. Shin, C. Lichtenwalner, M. Kozhevnikov, and J. Kim, "Perforamnc of an optical switch based on 3-D MEMS crossconnect," *IEEE Photon. Technol. Lett.*, vol. 16, no. 3, pp. 780–782, Mar. 2004.

[2] V. A. Aksyuk, S. ARney, N. R. Basavanhally, D. J. Bishop, C. A. Bolle, C. C. Chang, R. Frahm, A. Gasparyan, J. V. Gates, R. George, C. R. Giles, J. Kim, P. R. Koloner, T. M. Lee, D. T. Neilson, C. Nijander, C. J. Nuzman, M. Pacskowski, A. T. Papazian, F. Pardo, D. A. Ramsey, R. Ryf, R. E. Schotti, H. Shea, and M. E. Simon, "238 × 238 Micromechanical optical cross connect," *IEEE Photon. Technol. Lett.*, vol. 15, no. 4, pp. 587–589, Apr. 2003.

[3] V. Kaman, Z. Zheng, R. J. Helkey, C. Pusarla, and J. E. Bowers, "A 32-element 8-bit photonic true-time delay system base on a 288 × 288 3-D MEMS optical switch," *IEEE Photon. Technol. Lett.*, vol. 15, pp. 849–851, Jun. 2003.

[4] X. Zheng, V. Kaman, S. Yuan, Y. Xu, O. Jerphagnon, A. Keating, R. C. Anderson, H. N. Poulsen, B. Liu, J. R. Sechrist, C. Pusarla, R. Helkey, D. J. Blumenthal, and J. E. Bowers, "Three-dimensional MEMS photonic cross-connect switch design and performance," *IEEE J. Sel. Topics Quantum Electron.*, vol. 9, no. 2, pp. 571–577, Mar./Apr. 2003.

[5] V. Argueta-Diaz and B. L. Anderson, "Reconfigurable photonic switch based on a binary system using the white cell and micromirror arrays," *IEEE J. Sel. Topics Quantum Electron.*, vol. 9, no. 2, pp. 594–602, Mar./Apr. 2003.

[6] B. L. Anderson, V. Argueta-Diaz, F. Abou-Galala, G. Radhakrishnan, and R. Higgins, "Optical cross-connect switch based on Tip/Tilt micromirrors in a white cell," *IEEE J. Sel. Topics Quantum Electron.: Opt. Interconnects*, vol. 9, no. 2, pp. 579–593, Mar./Apr. 2003.

[7] D. J. Rabb, O. Blum-Spahn, W. D. Cowan, and B. L. Anderson, "Optical Fourier cell for true time delay," *J. Lightw. Technol.*, vol. 27, no. 7, pp. 879–886, Apr. 2009.

[8] J. W. Goodman*, Introduction to Fourier Optics*. San Francisco, CA: McGraw-Hill, 1968.

David J. Rabb received the B.S. degree in electrical engineering from Ohio University, Athens, in 2002, and the M.S. and Ph.D. degrees in electrical and computer engineering from the Ohio State University, Columbus, in 2005 and 2008, respectively.

He is currently with the Electro-Optic Sensors Division, Air Force Research Laboratory, Wright-Patterson Air Force Base, Dayton, OH. His current research interests are optical signal processing, lens design for imaging and nonimaging systems, laser radar, and optical communications.

Betty Lise Anderson (S'75–M'79–SM'95) received the B.S. degree in electrical engineering from Syracuse University, Syracuse, NY, and the M.S. and Ph.D. degrees in materials science and electrical engineering from the University of Vermont, Burlington, in 1978, 1988, and 1990, respectively.

She spent nine years in industrial companies including Tektronix, Inc., GTE Laboratories, and Draper Laboratories. She is currently a Professor at the Department of Electrical and Computer Engineering, The Ohio State University, Columbus. Her current research interests include analog optical signal processing, devices for optical communication systems, coherence, and semiconductor devices. She is the coauthor (with R. L. Anderson) of *Fundamentals of Semiconductor Devices* (McGraw-Hill, 2005).

Prof. Anderson is a member of the Optical Society of America. She is an Associate Editor for IEEE Journal of Quantum Electronics.