Arrayed Waveguide Grating

Dr. Martin Amersfoort
Concept to Volume b.v.

15 June 1998
Application note A1998003

Table of Contents

Abstract

This application note requires the basic and advanced mode solver modules in the Selene/Cross-section interface, and the BPM and mask layout modules in the Prometheus/Topview interface.

This application note describes how to design, simulate and layout an Arrayed Waveguide Grating (AWG) using **OlympIOs**.
The operation principle of the AWG is described and additionally some simple design rules are given.
We have described how to simulate the AWG by means of Beam Propagation Methods (BPM), by breaking it into three sections: an input star, an array of decoupled waveguides and an output star. A simulation example is presented which shows that the relevant parameters related to the AWG's spectral response can be obtained using this method. Moreover, they agree quite well with the design values.
Finally, we have provided the details concerning the implementation of these designs into **OlympIOs**, using the hierarchical design features incorporated in this package.

1. Introduction

In recent years Arrayed Waveguide Gratings ([Smit88], [Taka90], [Drag91]) have become increasingly popular as wavelength (de)multiplexers for WDM applications. They have proven to be capable of precise demultiplexing of a large number of channels with relative low lossses.

This application note describes how to design and simulate an Arrayed Waveguide Grating (AWG) using **OlympIOs**. The contents of this note are as follows.

General AWG design, page 3, outlines the operation principle of the AWG and in addition gives some simple design rules for this component. BPM Simulation, page 10, describes the simulation of the AWG using the Beam Propagation Method (BPM). Layout and simulation example, page 13, describes the results of a design and simulation of an example design. Implementation details, page 17, describes some tips with respect to the implementation of the AWG design into **OlympIOs**.

2. General AWG design

This chapter describes the basic operation principles of the AWG (also known as the optical phased-array), how to calculate the basic properties such as dispersion, and provides some simple design rules for optimum performance.

2.1. Operation principle

Figure 1 shows a schematic representation of the phased-array demultiplexer. The device consist of two slab waveguide star couplers, connected by a dispersive waveguide array. The operation principle is as follows. Light propagating in the input waveguide will be coupled into the array via the first star coupler. The array has been designed such that (for the central wavelength of the demultiplexer) the optical path length difference between adjacent array arms equals an integer multiple of the central wavelength of the demultiplexer. As a consequence, the field distribution at the input aperture will be reproduced at the output aperture. Therefore, at this wavelength, the light will focus in the center of the image plane (provided that the input waveguide is centered in the input plane).

(a)

Fig. b

(b)

d

f

$d\Theta$

y
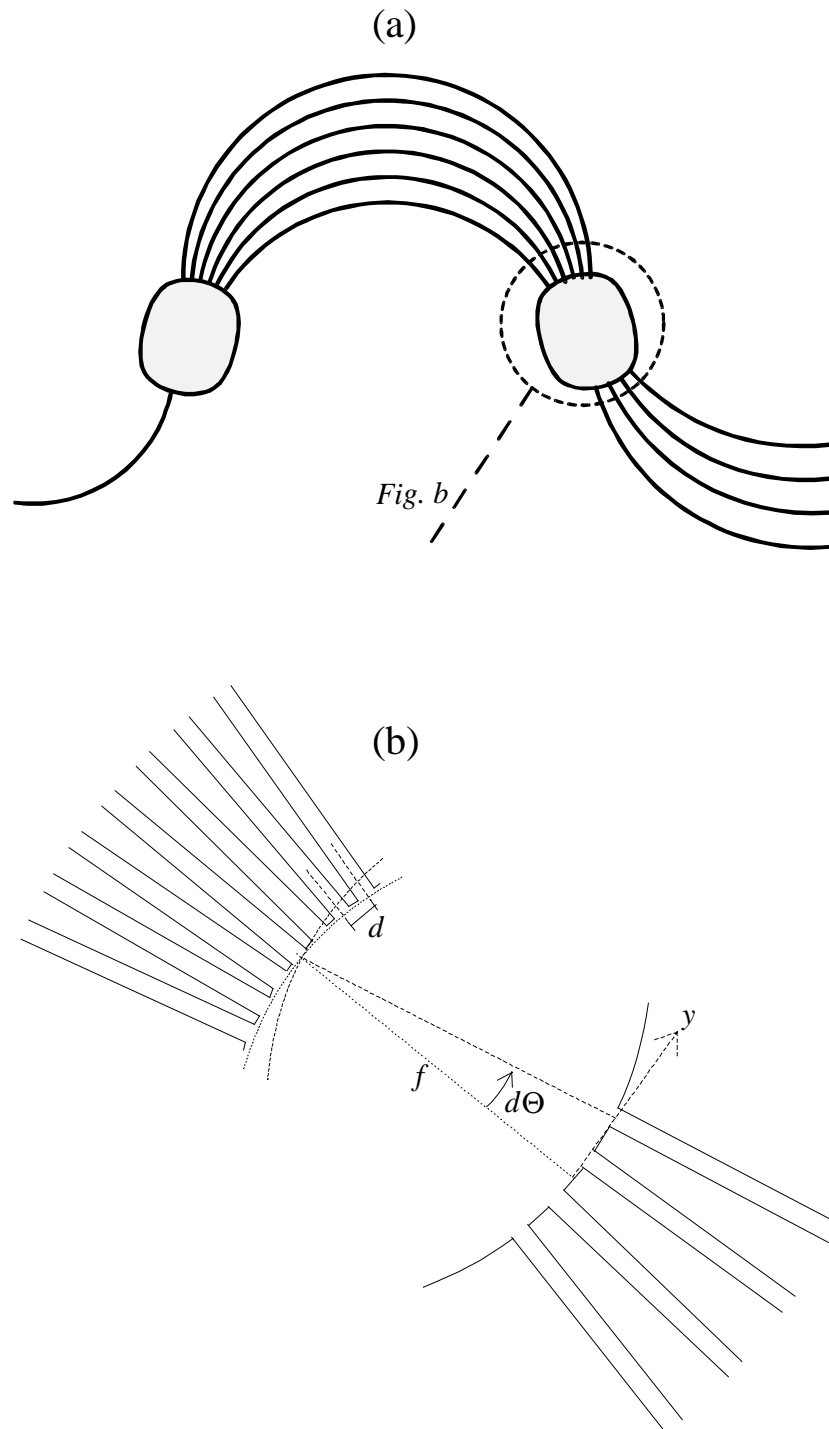
Figure 1. **Schematic representation of the phased-array demultiplexer**

(a) and magnification of the output star coupler (b)

If the input wavelength is detuned from this central wavelength, phase changes will occur in the array branches. Due to the constant path length difference between adjacent waveguides, this phase change will increase linearly from the inner to outer array waveguides, which will cause the wavefront to be tilted at the output aperture. Consequently, the focal point in the image plane will be shifted away from the center. The positioning of the output waveguides in the image plane allows the spatial separation of the different wavelengths (or frequencies).

2.2. Dispersion

The wavelength dependent shift of the focal point in the image plane can be calculated as follows. Using the parameters defined in Figure 1.b, the wavefront tilting angle $d\theta$ due to a phase difference $d\phi$ between adjacent array waveguides can be expressed as:

$$d\theta \quad = \quad \arcsin\left(\frac{d\phi / \beta_f}{d_g}\right) \quad \approx \quad \frac{d\phi / \beta_f}{d_g} \tag{1}$$

with β_f the propagation constant in the slab waveguide and

$$d\phi = d\beta_g \Delta l = \beta_g (dN_{eff} / N_{eff} - d\lambda_c / \lambda_c)\Delta l \tag{2}$$

in which β_g and N_{eff} are the propagation constant and effective index of the array waveguide respectively, Δl is the path length difference between adjacent array arms and λ_c is the center wavelength of the phasar. Combining these equations gives

$$\frac{d\Phi}{d\lambda_c} = -\frac{rm}{N_{eff}d_g} \tag{3}$$

with $r = \beta_g/\beta_f \approx 1$ and m the diffraction order of the demultiplexer defined as

$$m = \frac{\Delta l}{\lambda_c / N_{eff}} \tag{4}$$

Note that for the sake of simplicity the material dispersion has been ignored in these expressions, though incorporation of this effect is straightforward.

Using the wavelength measured in the material $\lambda_g = \lambda_c/N_{eff}$ the wavefront titling can be simplified to

$$\frac{d\Phi}{d\lambda_g} = -\frac{rm}{d_g} \qquad ; \qquad m = \frac{\Delta l}{\lambda_g} \tag{5}$$

Finally, with $dy = f\, d\Phi$ the relative dispersion $\delta y = \lambda_g(dy/d\lambda_g)$ of the focused spot in the image plane is easily determined to be:

$$\delta y = \lambda_g \frac{dy}{d\lambda_g} = -r \frac{mf\lambda_g}{d_g} \tag{6}$$

with f being the focal length of the demultiplexer.

2.3. Free spectral range

Another important property of the AWG is the Free Spectral Range (FSR), also known as the demultiplexer periodicity. This periodicity is due to the fact that constructive interference at the output star coupler can occur for a number of wavelengths and is easily calculated by specifying

$$\Delta l = m\frac{\lambda_c}{N_{eff}(\lambda_c)} = (m-1)\frac{\lambda_c + \Delta\lambda_{FSR}}{N_{eff}(\lambda_c + \Delta\lambda_{FSR})} \tag{7}$$

After some manipulations (and again ignoring material dispersion) we end up with

$$\Delta\lambda_{FSR} \approx \frac{\lambda_c}{m} \tag{8}$$

2.4. Branch geometry

So far, we have not yet discussed the exact geometry of the array branches. Obviously one can imagine an infinite number of ways to design the array branch. One of the most practical ways to construct the array branch (as proposed by [Smit91]), that leads to a rather compact design, is depicted in Figure 2. It consists of a (non-concentric) curved waveguide (of adjustable radius of curvature) smoothly connected to a straight waveguide (of adjustable length) on either side of the curve.
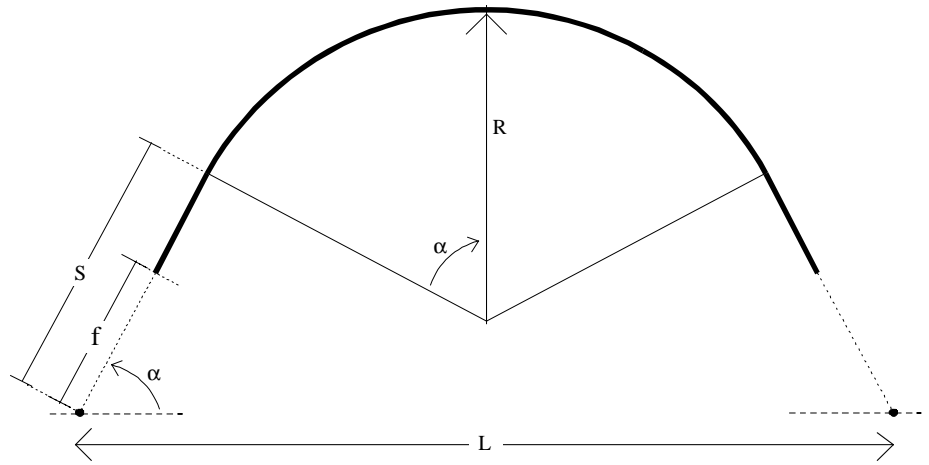
Figure 1. Schematic representation of the array branch geometry

The straight section length S_i (including the focal length f, for ease of calculation) and radius of curvature R_i for the i-th array guide can easily be calculated to be

$$\alpha_i = \alpha_1 + (i-1)\Delta\alpha \qquad ; \qquad i = 1, n_g \tag{9}$$

$$S_i = 1/2\left(l_i - \frac{\alpha_i L}{\sin(\alpha_i)}\right)\Big/\left(1 - \frac{\alpha_i \cos(\alpha_i)}{\sin(\alpha_i)}\right) \tag{10}$$

$$R_i = \frac{(1/2)L - S_i \cos(\alpha_i)}{\sin(\alpha_i)} \tag{11}$$

in which l_i is the path length of the i-th element, which has to satisfy the condition $l_i = l_1 + m(i-1)\lambda_c$.

Obviously there are still several degrees of freedom in this design. These can be used to optimize the array geometry.

2.5. Design strategy

Based on the properties calculated above a simple design strategy can be easily devised and is described below. A more elaborate discussion of AWG design aspects can be found in [Smit96].

The first step is to fix the spacing of the output waveguides in the image plane. In order to achieve sufficient isolation between neighbor output waveguides the gap between the output waveguides should be sufficiently large. As a rule of thumb, this gap should be chosen to be twice the width of the waveguide. Now the output waveguide spacing has been fixed the relative dispersion δy can be calculated to be

$$\delta y = d_{ch} / \left(\Delta\lambda / \lambda_c \right) \tag{12}$$

with $\Delta\lambda$ the channel spacing of the demultiplexer

For a fixed Free Spectral Range, the array order m can be calculated , using expression (8), to be

$$m = round\left(\frac{\lambda_c}{\Delta\lambda_{FSR}} \right) \tag{13}$$

The rounding of to the nearest integer is necessary to fix the center wavelength to the specified value. Note that this will lead to a slight correction of the FSR.

Now that the array order has been fixed, the focal length f can be calculated, using expression (6), to be

$$f = \frac{d_g \delta y}{rm\lambda_g} \tag{14}$$

The angular increment $\Delta\alpha$ of the array branch angle simply follows as

$$\Delta\alpha = d_g \, / \, f \qquad\qquad (15)$$

The last thing that needs to be done is to take care that the path length difference of the waveguide array branches is set correctly. Using equations (5), (6) and (15) the first derivative of the path length $dl/d\alpha$ $(= \Delta l/\Delta\alpha)$ can expressed as

$$\frac{dl}{d\alpha} = \frac{\delta y}{r} \qquad\qquad (16)$$

Therefore, the element length can be expressed as a function of the input angle α

$$l(\alpha) = l_0 + (\delta y \, / \, r)\alpha \qquad\qquad (17)$$

with l_0 a design parameter for further optimization.

The current design still has several degrees of freedom, which may be used to optimize the design. At this stage the input to output plane spacing L and the value of l_0 are still to be determined. Selection of S_r and R_r at a reference angle α_r fixes them to

$$L = 2S_r\cos(\alpha_r) + 2R_r\sin(\alpha_r) \qquad\qquad (18)$$

$$l_0 = l_r - (\alpha_r \, / \, r)\delta y = 2S_r + 2\alpha_r R_r - (\alpha_r \, / \, r)\delta y \qquad\qquad (19)$$

Several optimization procedures have been devised in order to arrive to an optimum layout ([Smit91], [Amer94]). A detailed discussion of these is considered, however, beyond the scope of this application note.

The last thing that needs to be decided is the number of array waveguides. This number should be sufficiently large, such that almost all the light diffracted into the free space region is collected by the array aperture. As a rule of thumb this number should be bigger that four times the number of wavelength channels.

3. BPM Simulation

This chapter describes how simulate the performance of a phased-array demultiplexer using Beam Propagation Methods (BPM). Since these simulations are quite CPU intensive we recommend to use a high-end PC such as a PentiumPro.

3.1. General principle

The basic idea behind this simulation method is to break the AWG into three parts: the input star coupler, an array of decoupled waveguides, and the output star coupler. First the input star coupler is simulated with BPM (For the central wavelength of the AWG only! Note that this can be done since the star coupler performance is very insensitive to wavelength.). For each of the array waveguides the power and phase are determined at a location where the waveguides are sufficiently decoupled. Subsequently the phase change in each array waveguide $(\beta_g l_i)$ is calculated by taken into account the optical path length of waveguide. Finally the second star coupler is simulated with BPM as a function of wavelength. The startfield for this simulation consist of the eigenmodes of each of the array waveguides, taken into account the proper power and phase.

3.2. Simulation of the input star

Figure 3 shows a schematic representation of the configuration for the simulation of the first star coupler. The eigenmode of the input waveguide is used as the startfield of this simulation.

Overlap elements are positioned on all the array waveguides in order to determine the power and phase in each of the waveguides.

It is recommended to put the overlap elements on a straight line, contrary to initial intuition. This issue is related to the BPM algorithms implemented in **OlympIOs**. These algorithms, based on the 'Slowly Varying Envelope Approximation' [Hoek93], calculate field changes with respect to the field change that would occur in a reference medium with index n_{ref}.
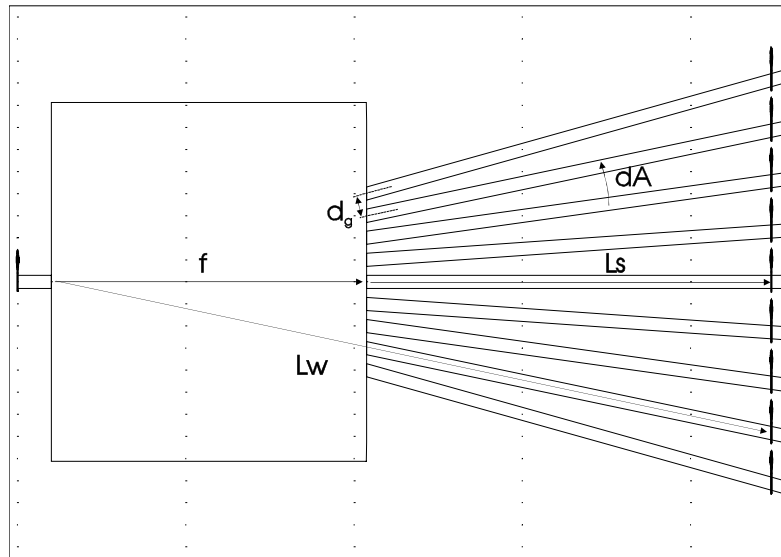
**Figure 3. Schematic representation of the
input star coupler simulation configuration**

The phase information in the program is also stored with respect to this reference index. Consequently it is rather tricky to determine phase differences between points that are located at different z-coordinates.
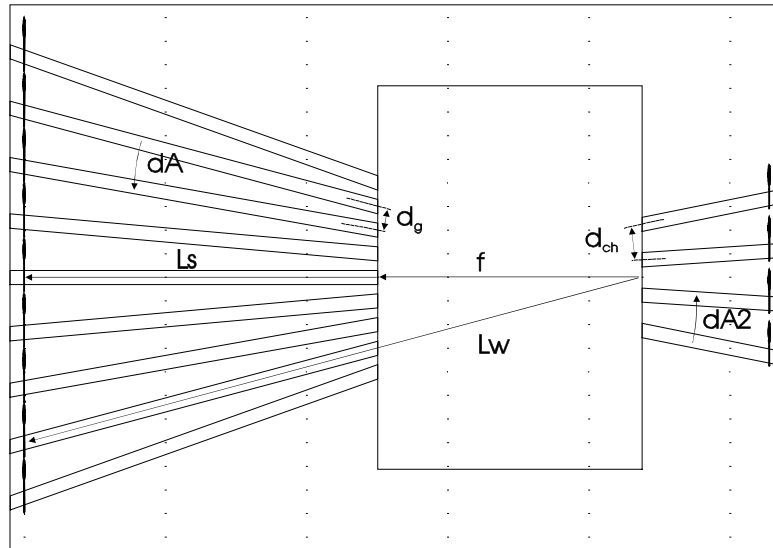
3.3. Simulation of waveguide array

The dispersive effect of the waveguide array is determined by calculating the phase change $\beta_g\, l_i$ as a function of wavelength. At the present time the propagation constant in the curved waveguide is assumed to be identical to the straight waveguide propagation constant. The expression for those phase changes can be easily incorporated into the DEV-file of the output star coupler. In order to allow this treatment, it is obviously essential that the waveguides are sufficiently decoupled at the end of the input star coupler simulation region.

3.4. Simulation of the output star coupler

The output star coupler is simulated again with the BPM method (see Figure 4). The startfield consists of the eigenmodes of the array waveguides with the proper power and phase relations. It is

recommended, again, to position the modal start field elements at the same z-coordinate, to avoid phase complications. The phase correction with respect to the position on a circle can be easily calculated to be:

$\beta_c dL$ *(input star)* $+ \beta_d dL$ *(output star),* with $dL = Lw - (f+Ls)$.

Figure 4. Schematic representation of the simulation configuration for the output star coupler

Overlap elements are positioned on the output waveguides to analyze the power in each of the output channels of the AWG. The spectral response can be calculated by performing a "vary BPM" with the wavelength as variable. Be aware that a full wavelength sweep will typically take a few hours on a PentiumPro.

3.5.   Algorithm and discretization

In order to obtain reliable simulation results it is important to select the proper algorithm and discretization. Generally, it is recommended to use the second or fourth order Padé FD algorithm (FD2 or FD4) for these simulations. Although the power loss of angled waveguides may seem small for the FD0 algorithm, the phases can be significantly off. It was found that the FD0 algorithm only produces reliable simulation results for very low-contrast large-width waveguides.

Even if the FD2 algorithm will be used, care has to be taken to choose the right propagation step size and number of grid-points in the x-direction. If the number of points is too small (in x and/or z-direction), the interference pattern in the star coupler may still seem to be realistic, but can be slightly defocused, which will overestimate the AWG insertion loss.

4. Layout and simulation example

In order to illustrate the design and simulation concepts outlined in the previous chapters, an example is presented for an AWG with these external parameters:

Variable	Value
n_1	1.455
n_o	1.450
λ_c	1.55 µm
$\Delta\lambda$	3.2 nm
w_g	6 µm
n_{ch}	8
d_g	15 µm
w_t	12 µm
d_{ch}	20 µm
$\Delta\lambda_{FSR}$	25 nm
L	20 mm
l_0	16.4 mm
D_{out}	250 µm

with n_1 and n_0 the effective index of the center and outer waveguide region, w_t the width of the taper at the input and output aperture and D_{out} the spacing between the input and output waveguides.

4.1. Layout of the phasar

Based on the simple design rules provided in the previous chapters, the layout displayed in Figure 5 has been obtained. The specific choice of

L and l_0 guarantees that the minimum radius of curvature is larger than 10 mm. The die size is 50×12 mm^2 for this example.
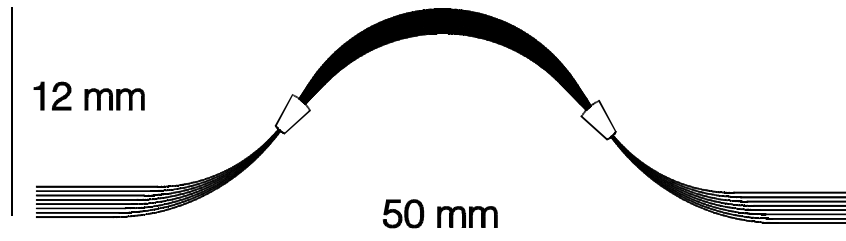
Figure 5. Layout of an 8-channel AWG demultiplexer

Figure 6 shows a detail of the input star coupler, with the characteristic Rowland circle positioning of the input waveguides.
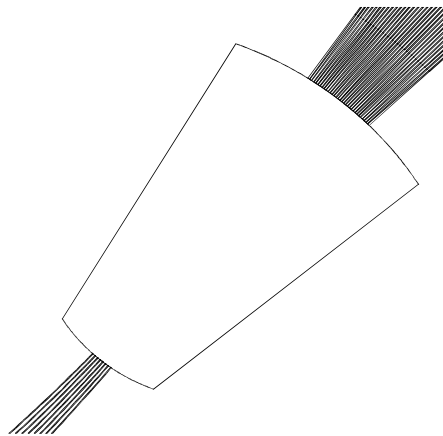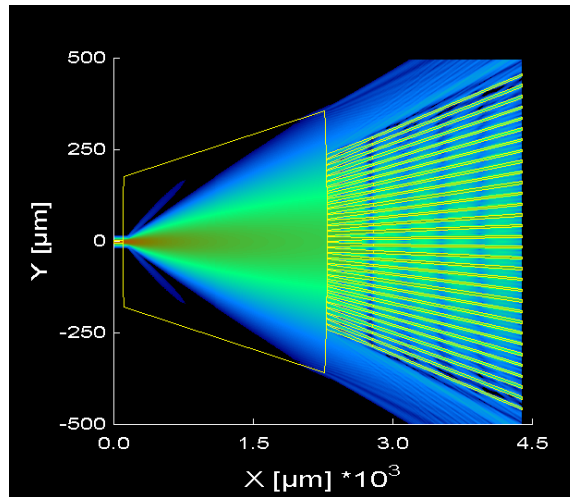

Figure 6. Detail of the input star of the AWG demultiplexer

4.2. Single wavelength simulation

Figure 7. shows the intensity distribution (log-scale) of the input star simulation of the example specified above. We used the FD2 (wide angle) propagation algorithm with 4096 points in the x-direction and a propagation step size of 4 µm.
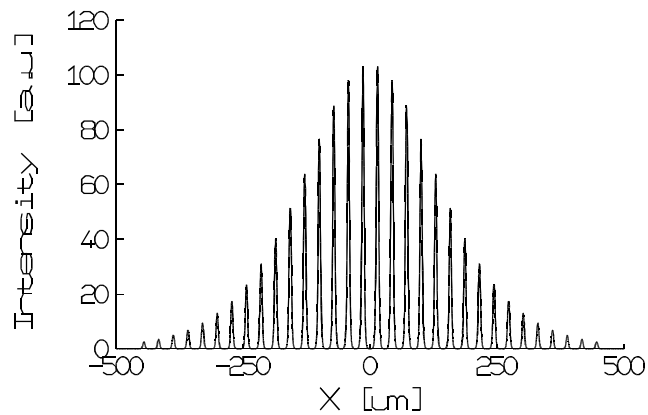
Figure 7. Field intensity distribution (log-scale)
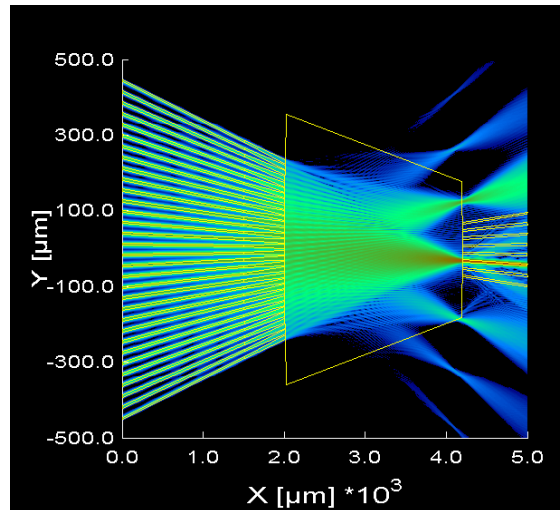
of the output star coupler simulation

The total power in the guided modes of the array waveguides at the end of the input star was found to be 84.6 %.

Figure 8. shows the startfield of the output star simulation. This startfield is the superposition of the eigenmodes of the individual array waveguides with the proper power and phase correction.
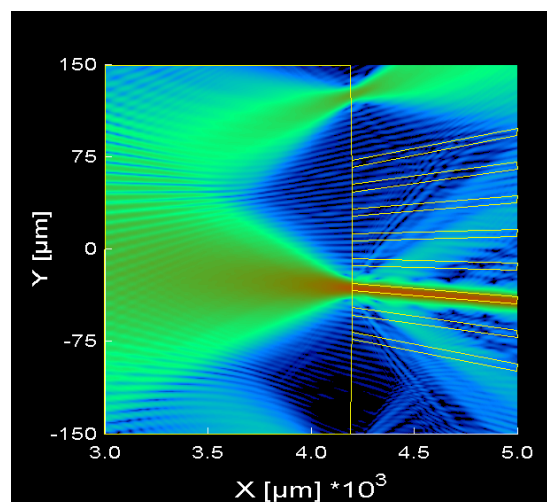

Figure 8. Field intensity distribution of the start field

for the output star coupler simulation

Figure 9. shows the field intensity distribution (log-scale) for a wavelength of 1554.8 nm. Proper focusing of the output beam in the image plane can be observed. The adjacent diffraction orders are also clearly visible.

Figure 9. Field intensity distribution (log-scale) of the output star coupler simulation. For the black regions in these intensity plots the signal level is more than -60 dB down

A detailed magnification of the image plane (Figure 10.) clearly shows that the spot couples into output number 6, as would expected based on the center wavelength of 1550 nm and channel spacing of 3.2 nm.


Figure 10. Detail of the intensity distribution (log-scale) in the image plane of the output star

The total power (as a fraction of the start field power) in the output waveguide was found to be 87.4%. This amounts to a total insertion loss of -10*log(0.846*0.874)= 1.3 dB.

4.3. Wavelength sweep

Finally the spectral response was calculated by performing a vary-run versus the wavelength. The wavelength step was chosen to be 0.2 nm. Figure 11. shows the calculated spectral response for this structure. The insertion loss ranges from 1.0 dB for the center to 3.5 dB for the outer channel. The crosstalk is found to be better than -40 dB.
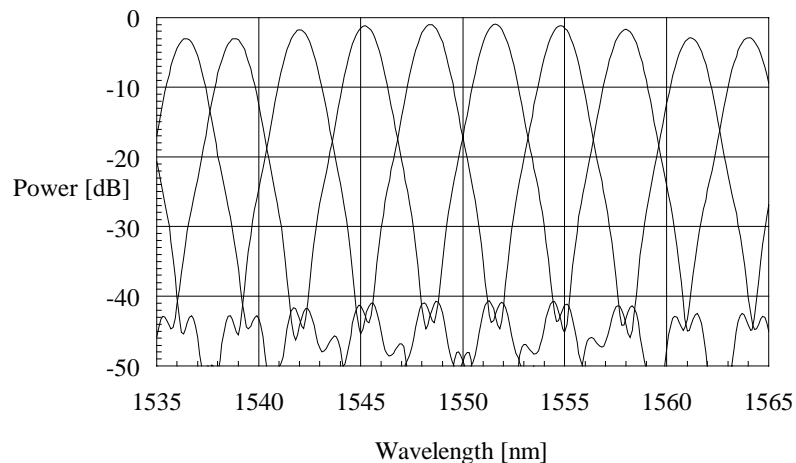
Figure 11. Calculated spectral response of the simulation example. notice the periodicity of the demultiplexer

5. Implementation details

In this chapter we describe some hints concerning the implementation of the AWG design described in the previous chapters in the **OlympIOs** design software. Example files have been included with the software to facilitate the understanding and creation of these designs.

5.1. General considerations

When implementing a complex component like an phased array demultiplexer in **OlympIOs** it is recommended to parameterize the design as much as can be reasonably achieved within the possibilities of the DEV-format. Specifically use of the following features/functions of **OlympIOs** is necessary to implement the design strategy presented in previous chapters (for a detailed description we refer to the **OlympIOs** user manual).

- Variables and expressions
- Group definitions (to structure the design)
- For loops (for example to layout the phasar)
- Conditional statements
- Assertions
- Info text statements (to view calculated results, or to debug the design).
- List document statements (to improve readability of the 'Structure list').

The most relevant AWG parameters have been entered by means of an include file. This file is also shared by the DEV-files for the input and output star coupler BPM simulation. If these parameters are changed, they will be automatically updated in all three files. This way we can make sure that the layout and simulated structure are identical.

5.2. Layout of the phasar

The structure is defined by a number of group calls (some of them within for loops). The 'For Branch_angle' loop defines the waveguide array. The 'For Out_angle' loop defines the output waveguides and the 'For In_angle' loop defines the input waveguides (see Figure 12). Finally the free propagation regions (Rowland circles) are defined using a curve statement.

Two dummy layers have been added, that should not be used for the final mask export. The purpose of these layer 'dummy1' is to visualize that the array all originate and terminate in the same point. The purpose of layer 'dummy2' id to visualize that the input (output) branches are directed towards the center of the input (output) aperture.
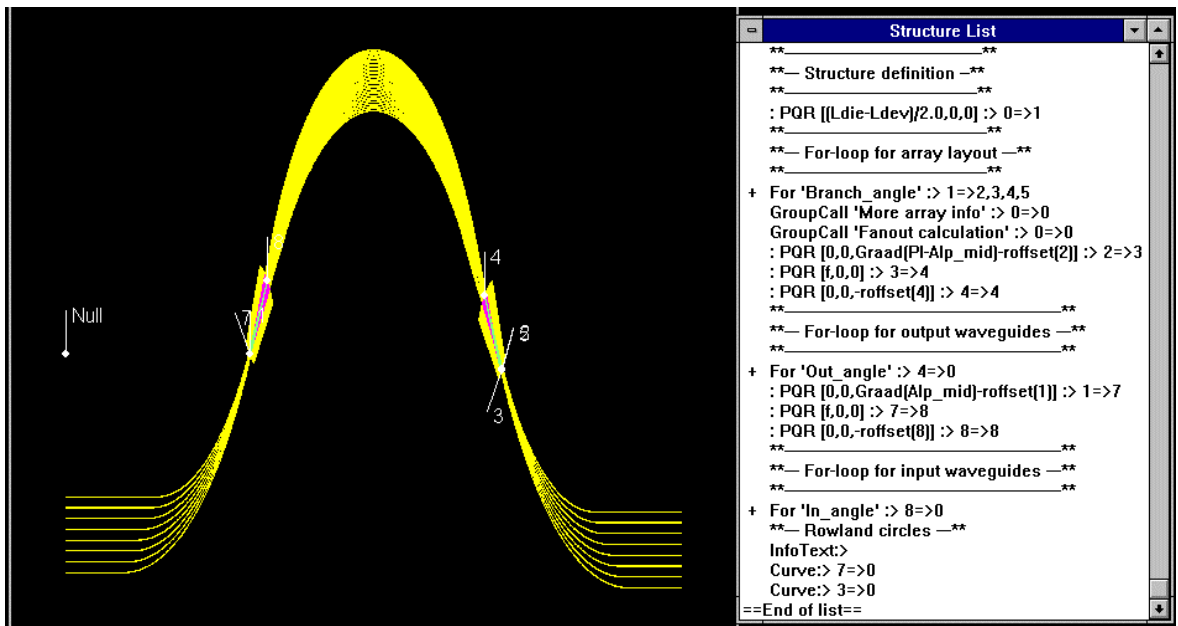
**Figure 12. Total layout of the AWG and corresponding section
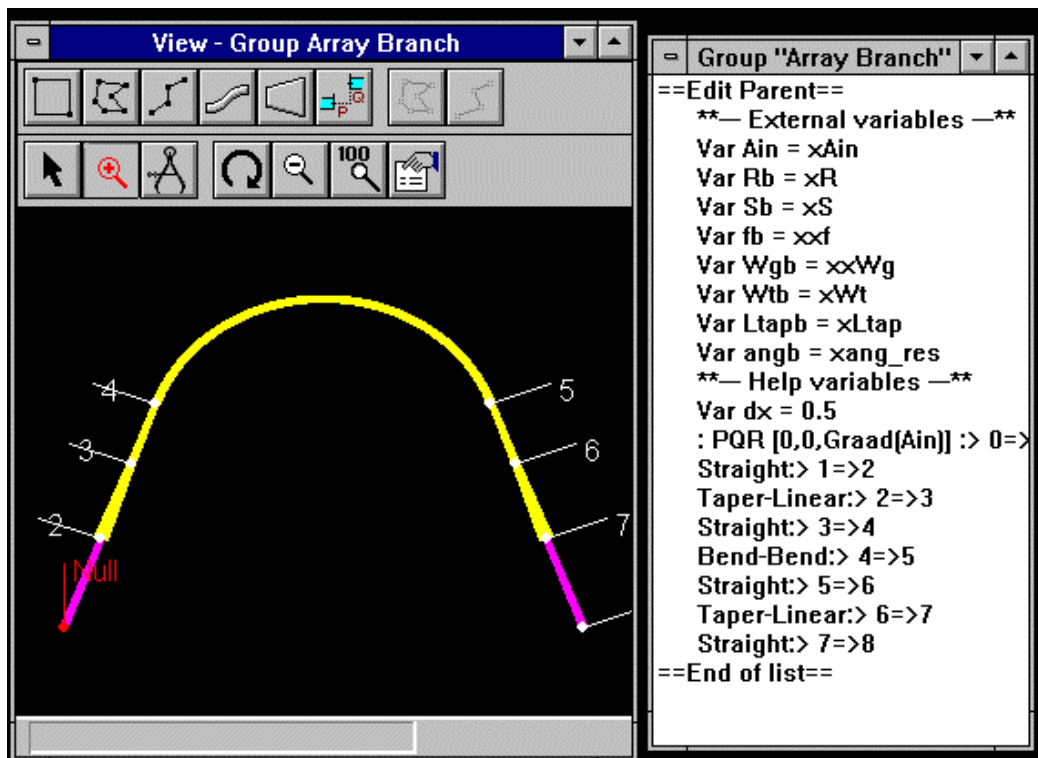of the 'Structure list'**

Figure 13. Layout and 'Structure list' of the 'Group Array branch'

The following sub-groups have been used:

- **Array branch:** Performs the layout of a single array branch (see Figure 13).
- **Output branch:** Performs the layout of a single output branch (see Figure 14).
- **Input branch:** Performs the layout of a single input branch
- **Fanout calculation:** Calculates the geometry of the innermost input/output branch
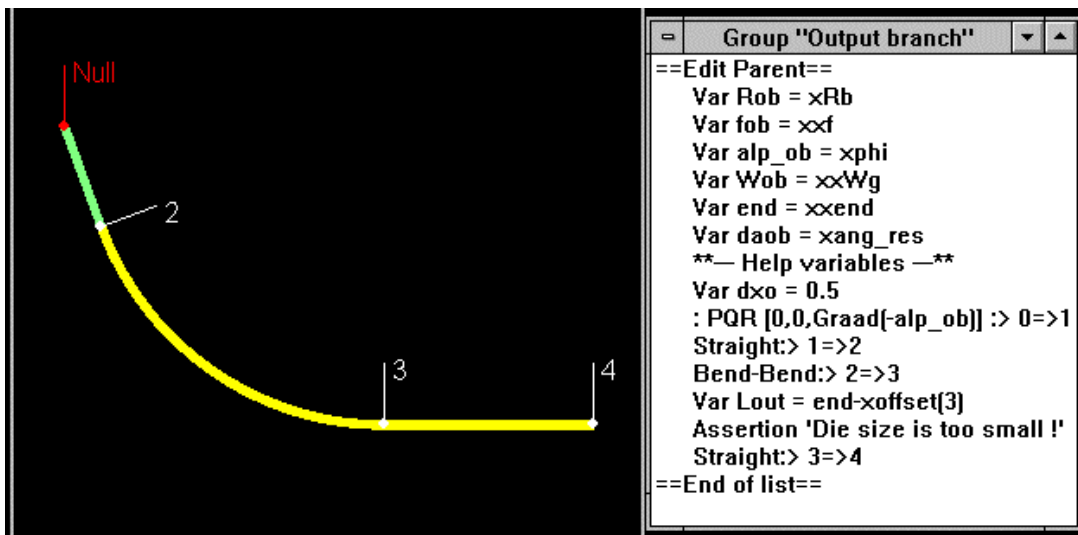- **More array info:** Displays some additional information of the array geometry in the Expansion-window.

Figure 14. Layout and description of the 'Group Output Branch'

5.3. Simulation of the phasar

The DEV-files of the input and output star simulation are pretty much self-explaining. Several list comments have been added in order to explain what has been done. The most tricky part is to enter the proper phase and power of the first simulation into the second BPM simulation. This has been done using the External element ***ReadRp2Plane***. This element reads the power and phase values of a simulation results file (*.rp) to be specified in the predefined variables "Pow" and "Phase_in". Therefore the way to connect the input and

output star simulation is by making sure the results file (*.rp) is saved after completion of the input star simulation and specifying the same filename in the **ReadRp2Plane** element (within the For-loop "FordA") of the output star simulation. Info text statements have been added to buffer the power and phase relations to the Info-window, in order to check that they were entered correctly.

NOTE: Where the **ReadRp2Plane** external element is not included in the GUI, it is quite likely the "c:\bbv\extern.reg" file does not include a reference to the executable "c_rppln.exe". Adding external elements to the standard GUI is described in detail in the Mask layout Module, "Optical element library" of **OlympIOs** see "External Elements". The "standard" **C2V OlympIOs** Software external elements, including the **ReadRp2Plane** element, can be included in the interface by replacing the current "extern.reg" file with the file named "extern.reg-example" in the "External elements subdirectory".

6. References

[Amer94]	M.R. Amersfoort, 'Phased-array wavelength demultiplexers and their integration with photodetectors', *Ph.D. thesis*, Delft University of Technology, ISBN 90-407-1041-4, 1994
[Drag91]	C. Dragone, 'An N×N optical multiplexer using a planar arrangement of two star couplers', *IEEE Phot. Techn. Lett.*, **3** (9), pp. 812-815, 1991
[Hoek93]	H. Hoekstra et al., 'New formulation of the beam propagation method based on the slowly varying envelope approximation', Opt. Comm., 97, pp. 301-303, 1993
[Smit88]	M.K. Smit, 'New focusing and dispersive component based on an optical phased-array.', *Electron. Lett.*, **24** (7), pp. 385-386, 1988
[Smit91]	M.K. Smit, 'Integrated Optics in silicon-based aluminum oxide', Ph.D. Thesis, Delft University of Technology, ISBN 90-9004261-X, 1991
[Smit96]	M.K. Smit and C. van Dam, "PHASAR-Based WDM-Devices: Principles, Design and Applications", *IEEE J. of Sel. Topics in QE*, **2** (2), pp. 236-250, 1996
[Taka90]	H. Takahashi *et al.*, 'Arrayed waveguide grating for wavelength division multi/demultiplexer with nanometre resolution', *Electron. Lett.*, **26** (2), pp. 87-88, 1990

About C2V Application Notes

The purpose of **C2V** application notes is two-fold: They serve as a platform of discussion for integrated-optic design and simulation issues and problems. And they address the specific details of the design implementation and/or simulation problem to **C2V**. Example files are generally included, in order to give the customer a head start in his/her design effort.

Application notes are distributed with our software products. Or they can be downloaded from our web site. Order lists for hard copies can be obtained upon request.

We encourage users of our software to write application notes about their own design and simulation problems that might be of interest to other integrated optic design engineers. Such an application note could, for example, describe how a specific component (for example a DOS-switch) is designed and simulated. Ideally, it should contain tips and tricks that will be helpful to other design engineers or researchers working on similar problems. Each year the best application note will be rewarded with a free service contract for one year for one license of a **C2V** software product. Please send your notes (and optional example files) to the address listed below. A template for formatting the notes can be obtained upon request.

For further information, please contact:

Postal Address:	**C2V**	Visits and deliveries/shipments:
	P.O Box 318	Colosseum 20
	7500 AH Enschede	7521 PT Enschede
	The Netherlands	The Netherlands
Tel:	+31 53 4 889 889	
Fax:	+31 53 4 889 890	
E-mail:	software@c2v.nl	
Internet:	www.c2v.nl	