SYNOPSYS®

Simulation of Triple-Junction GaInP/GaAs/InGaAs Solar Cells with Transfer Matrix Method

Abstract

This template project demonstrates the use of the transfer matrix method of TCAD Sentaurus for the simulation of the optical and electrical characteristics of a triple-junction GaInP/GaAs/InGaAs solar cell. Reflectance and external quantum efficiency spectra, as well as the light current density–voltage (J–V) curves and the energy band diagram, are simulated.

In addition, this template project demonstrates the use of the Epi utility to create a multilayer stack and the use of the MatPar utility to create user-defined material parameter files using user-specified material property models.

Moreover, the template project references the *Epi/MatPar User Guide*. The following link opens the manual *Epi/MatPar User Guide*.

Version Information

This application note has been designed and verified using TCAD Sentaurus Version A-2008.09.

Running it with previous or future versions may possibly require minor adjustments.

Introduction

As a powerful, general, semiconductor processing and device simulation suite, TCAD Sentaurus can simulate various optoelectronic devices. A comprehensive set of carrier and heat transport models, combined with an extensive set of optical models implemented in Sentaurus Device, addresses the needs of the optoelectronic community to simulate devices including CMOS image sensors, charge-coupled devices, and solar cells.

The Epi utility, in conjunction with Sentaurus Structure Editor, provides a way to easily create an epitaxial multilayer stack by specifying the layer properties in a comma-separated value (CSV) file.

The MatPar utility further extends the capabilities of Sentaurus Device by providing a tool command language (Tcl)–based framework for implementing user-defined models and creating the main parameter file and user-defined material parameter files using these models [1]. This approach of creating user-defined parameter files provides additional control of the handling of the material parameters, especially those of compound semiconductors.

The Material Parameter Database of MatPar provides an extensive collection of material parameters and models for several materials, which can be used as is or as a baseline for custom calibration. The Material Parameter Database is open and extensible. The user can implement new models using Tcl.

The Material Parameter Plotter of MatPar, which is an Inspect-based utility, can be used to visualize the variation of material properties (which are critical for the device operation) as a function of physical variables such as temperature, doping, mole fraction, and the wavelength of incident light. For example, the diffusion lengths of electrons and holes, and the absorption coefficient are important properties that contribute to the efficiency of the solar cell. For a particular material, the diffusion lengths can be visualized as a function of doping concentration for various models in the Material Parameter Database. In addition, the Material Parameter Plotter can be used to visualize the absorption coefficient as a function of energy for various materials. For details, refer to the *Epi/MatPar User Guide*, Material Parameter Plotter on page 74 [1].

For the simulation of planar devices, the transfer matrix method (TMM) can be used to calculate the optical field in devices consisting of multiple materials. Here, this approach is applied to the simulation of optical and electrical characteristics of a triple-junction GaInP/GaAs/InGaAs solar cell.

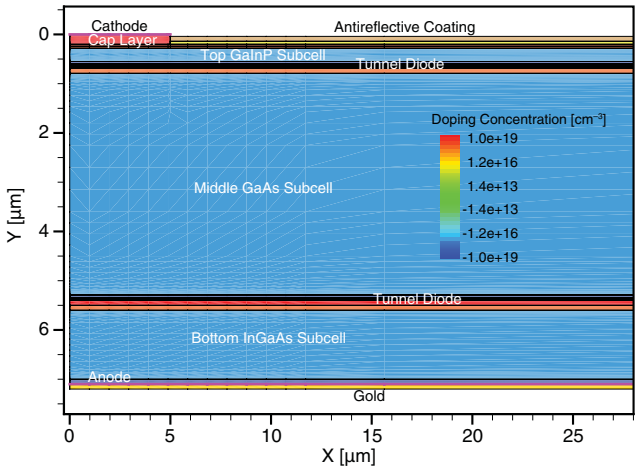

Figure 1 Smallest symmetry element of triple-junction solar cell generated by Sentaurus Structure Editor (used in 2D simulation mode)

The triple-junction GaInP/GaAs/InGaAs solar cell simulated here is a two-terminal monolithic tandem cell, and its structure is similar to that in the literature [2]. The structure is described in Table 1 on page 33 and is shown in Figure 1. It consists of three subcells and two tunnel diodes.

Each subcell is a single-junction solar cell of the same polarity (n/p). The top GaInP subcell is series connected to the middle GaAs subcell, and the middle GaAs subcell is series connected to the bottom InGaAs subcell through two p^+-AlGaAs/n^+-GaAs tunnel-diode interconnects.

The single-junction subcells and the tunnel diodes are components of the multijunction structure. Here, the electrical and optical characteristics of these components, as well as that of the complete triple-junction solar cell, are simulated.

The single-junction subcell structures of the multijunction solar cell are used to generate individual single-junction solar-cell structures, and the electrical and optical characteristics of these individual single-junction solar cells are simulated as well.

The electrical characteristics of the tunnel diode also is simulated. Accurate simulation of the electrical characteristics of the tunnel-diode interconnects is crucial for the successful simulation of triple-junction solar cells [3].

The layers in the monolithic structure comprise various materials such as AlInP, GaInP, AlGaAs, GaAs, and InGaAs. Here, the equilibrium energy band diagram of the AlInP/GaInP/AlGaAs/GaAs/InGaAs heterostructure and the tunnel diode are simulated. The energy band diagram of the individual single-junction solar cells, subcells, and triple-junction solar cell are also simulated under equilibrium and at short-circuit current density (J_{sc}).

Several standard solar spectra (AM0, AM1.5G, AM1.5D) [4] are included in this template project so that the light current density–voltage (J–V) curve can be simulated for various solar spectra.

All the solar-cell structures can be simulated either in 1D or 2D simulation mode. In 1D simulation mode, a homogenous 2D structure is created with only two or three vertical mesh lines. Simulations performed in 1D simulation mode are relatively faster and can be used to design the triple-junction solar cell quickly. The 1D simulation mode can also be useful for a faster optimization of the simulation mesh in the y-direction. The optimized y-direction mesh can be used in 2D simulation mode to optimize the mesh in the x-direction.

It is assumed that the user is familiar with the Sentaurus tool suite, in particular, Sentaurus Workbench, Sentaurus Structure Editor, Sentaurus Device, Inspect, and Tecplot SV. For an introduction and tutorials, refer to the Sentaurus training material.

The focus of this project is to provide a setup that can be used as is or adapted to specific needs. The documentation focuses on aspects of the setups. For details about tool uses and specific tool syntax, refer to the respective manuals.

This template project can be used to design III–V multijunction solar cells and the antireflective coating (ARC) used in these cells. It can be used to design the subcells for the current matching condition (the photocurrents of the subcells should be matched closely) [5].

Simulation Strategy and Results

This section discusses the simulation strategy used to simulate multijunction solar cells and the main simulation results.

The material parameters critical for solar cells, such as minority carrier mobility, minority carrier diffusion length, and absorption coefficient, are verified for the subcell n/p junction materials – $Ga_{0.5}In_{0.5}P$, GaAs, and $In_{0.3}Ga_{0.7}As$ – using the Material Parameter Plotter. The plots of these parameters are helpful in designing the triple-junction solar cell.

The band-structure material parameters of all the materials used in the triple-junction solar cell are verified by checking the equilibrium energy band diagram of the $Al_{0.5}In_{0.5}P/Ga_{0.5}In_{0.5}P/Al_{0.3}Ga_{0.7}As/GaAs/In_{0.3}Ga_{0.7}As$ heterostructure and by checking the band structure alignment in the heterostructure.

The tunnel diode simulation is used to optimize and verify the tunnel diode J–V characteristics. After optimizing the mesh using the equilibrium energy band diagram and selecting the tunneling parameters, the tunnel diode is designed so that the peak tunneling current density is greater than the J_{sc} of the triple-junction solar cell [6].

Before combining the subcells and tunnel diodes to create the triple-junction solar-cell structure, you verify that the individual subcells work properly by simulating the J–V characteristics and quantum efficiency (QE) spectra of the single-junction solar-cell structures created using the subcell structures.

Due to the optical shielding of the middle and bottom subcells in the tandem triple-junction structure, you must verify that the cells in the stack are current balanced at J_{sc}. The light J–V characteristics of each of these subcells also must be verified. This is performed using a dummy triple-junction solar-cell structure with four electrodes. The base-layer thickness in each subcell in the triple-junction solar cell is adjusted so that the J_{sc} of each subcell is within a few percent.

The optimized subcell base thicknesses are used in the single-junction simulations to quickly optimize the mesh in the y-direction (the growth direction of the layers) in the single-junction cells and, therefore, in the subcells. In addition, the optimized base thicknesses are used to verify the QE spectra of each subcell using the dummy triple-junction solar-cell structure. The QE spectra simulation can be used to further optimize the triple-junction structure by adjusting various parameters such as layer thicknesses and doping concentrations. The reflectance spectra simulation is used to design the ARC for the triple-junction solar cell.

After creating and optimizing all the components, the J–V curve of the triple-junction solar cell is simulated. It can be used to optimize the structure further.

Most of the optimization tasks in this simulation strategy can be performed first in the 1D simulation mode due to its relatively faster simulation time. The simulations can be repeated in the 2D simulation mode for accuracy.

Verifying Material Parameters and Band Structure

The absorption coefficient of all materials used in the triple-junction solar cell is plotted as a function of energy (see Figure 2 on page 5) using the `AbsorptionSpectra` instance of the Material Parameter Plotter in the `pardb/mprPlotter-MJSC` Sentaurus Workbench project. Figure 2 clearly shows the absorption edge of all materials used in the triple-junction solar cell.

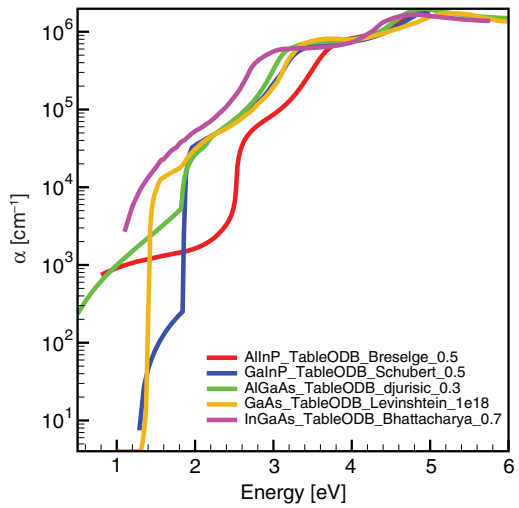

Figure 2 Absorption coefficient as a function of wavelength for $Al_{0.5}In_{0.5}P$, $Ga_{0.5}In_{0.5}P$, $Al_{0.3}Ga_{0.7}As$, GaAs, and $In_{0.3}Ga_{0.7}As$

The electron diffusion length in $Ga_{0.5}In_{0.5}P$, GaAs, and $In_{0.3}Ga_{0.7}As$ is plotted as a function of doping concentration (see Figure 3) using the `DiffusionLength` instance of the Material Parameter Plotter in the `pardb/mprPlotter-MJSC` Sentaurus Workbench project. Here, the electron diffusion length is calculated using the radiative recombination lifetime of the material. Figure 3 shows that the diffusion length decreases with increasing doping concentration for all materials.

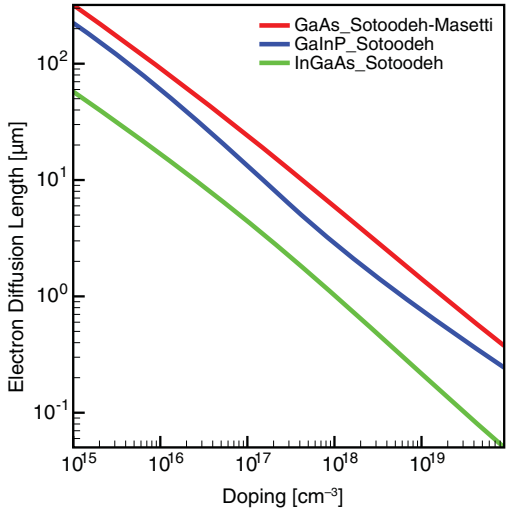

Figure 3 Electron diffusion length as a function of doping concentration in GaAs, $Ga_{0.5}In_{0.5}P$, and $In_{0.3}Ga_{0.7}As$

The `Mobility` instance of the Material Parameter Plotter plots the electron mobility in $Ga_{0.5}In_{0.5}P$, GaAs, and $In_{0.3}Ga_{0.7}As$.

The equilibrium energy band diagram of the $Al_{0.5}In_{0.5}P/Ga_{0.5}In_{0.5}P/Al_{0.3}Ga_{0.7}As/GaAs/In_{0.3}Ga_{0.7}As$ heterostructure (see Figure 13 on page 16 and Table 3 on page 34) is simulated in the Sentaurus Workbench project `01-Eqbm-Banddgm` and is shown in Figure 4. In this project, VBO is used to calculate the electron affinity value of each material with respect to the reference material for electron affinity, GaAs. Figure 4 is used to verify the VBO value of a material with respect to GaAs by measuring the difference in the value of the valence band edge (ΔE_V) of the material with respect to GaAs.

For example, in Figure 4, the ΔE_V of AlGaAs and InGaAs with respect to the reference material GaAs is –0.138 eV and 0.315 eV, respectively. For details about band offsets and the calculation of electron affinity, see Generating Parameter File Using MatPar on page 12.

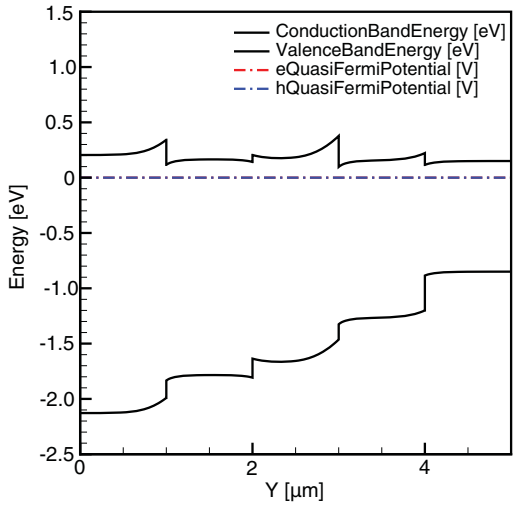

Figure 4 Equilibrium energy band diagram of $Al_{0.5}In_{0.5}P/Ga_{0.5}In_{0.5}P/Al_{0.3}Ga_{0.7}As/GaAs/In_{0.3}Ga_{0.7}As$ heterostructure

Verifying Tunnel Diode J–V Curve

The equilibrium energy band diagram (see Figure 5) and J–V characteristics (see Figure 6 on page 6) of the p^+-AlGaAs/n^+-GaAs tunnel diode (see Figure 14 on page 18) are simulated in the Sentaurus Workbench project `02-TunnelDiode-JV`. The mesh is optimized such that the gradients at the band edges in the equilibrium energy band diagram of the tunnel diode are resolved appropriately.

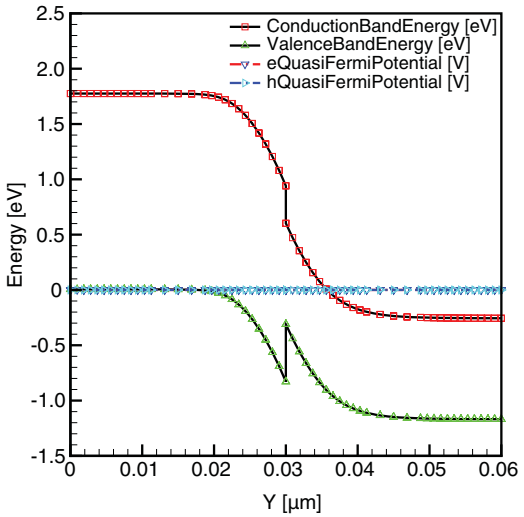

Figure 5 Equilibrium energy band diagram of AlGaAs/GaAs tunnel diode

The J–V curve in Figure 6 is a typical forward-biased J–V curve of a tunnel diode and exhibits a negative differential resistance region. At zero bias, the current is zero. As the forward-bias voltage increases, the current increases and reaches a peak value. With a further increase in the bias voltage, the current decreases to zero in the negative differential resistance region [7].

In Figure 6, the peak tunneling current density (J_{peak}) of the tunnel diode is 80.2 A/cm^2. As seen in Figure 10 on page 7, J_{peak} is much greater than the J_{sc} (12.31 mA/cm^2) of the triple-junction solar cell.

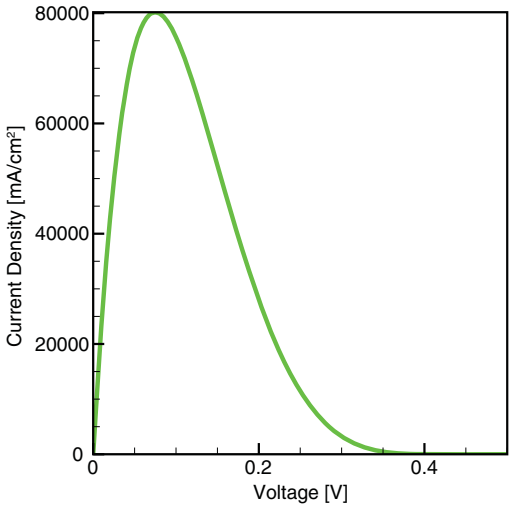

Figure 6 J–V characteristics of AlGaAs/GaAs tunnel diode

Verifying I–V and QE Spectra of Single-Junction Solar Cells and Optimizing Mesh

The Sentaurus Workbench project `03-SingleJn-JV-QE` simulates the light J–V and power density–voltage (P–V) curves, the QE spectra, and the spectral response (SR) of individual single-junction GaInP, GaAs, and InGaAs solar-cell structures. The subcell structures in the triple-junction solar cell are used to create these single-junction solar cells. The project also calculates the photovoltaic parameters of these cells. These quantities are defined in Inspect: Light J–V Curves on page 23 and Inspect: QE Spectra on page 24. The J–V, QE, and SR curves are used to verify that the individual subcell structures work properly.

Figure 7 shows the light J–V characteristics of the GaInP, GaAs, and InGaAs single-junction solar cells. As expected, the open-circuit voltage (V_{oc}) is the least for the InGaAs solar cell and the highest for the GaInP solar cell since V_{oc} depends on the band gap of the n/p junction material of the solar cell.

After the subcell base-layer thicknesses are optimized (see Optimizing Subcell Base-Layer Thickness), these base thicknesses are used in this project to optimize the y-direction mesh in the subcells and, therefore, the y-direction mesh in the triple-junction solar cell by resolving the gradients in the energy band diagram and the carrier concentration profiles.

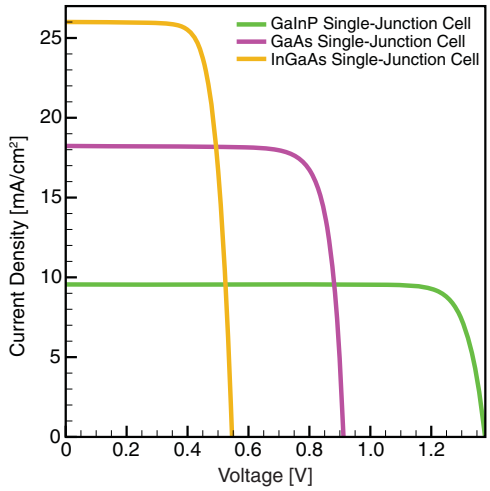

Figure 7 Light J–V characteristics of GaInP, GaAs, and InGaAs single-junction cells (2D simulation mode)

Optimizing Subcell Base-Layer Thickness

The Sentaurus Workbench project `04-MJSC-Subcell-Balance-Jsc` simulates the J_{sc} for each subcell in the triple-junction solar cell. In addition to plotting the energy band diagram at J_{sc} of the dummy triple-junction solar-cell structure, the Tecplot SV node in this project can be used to plot the optical generation profile for the triple-junction structure (see Figure 8).

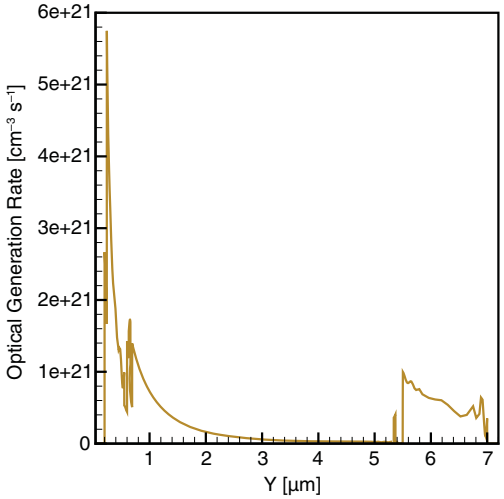

Figure 8 Optical generation rate profile for GaInP/GaAs/InGaAs triple-junction solar cell

The photocurrent and J_{sc} produced by each subcell are proportional to the integrated optical generation rate in the subcell. The subcells are current matched at J_{sc} by varying the base thicknesses of each subcell so that the integrated optical generation rate in each subcell is approximately the same. The optimized base thicknesses of the top GaInP subcell, middle GaAs subcell, and bottom GaInP subcell are 0.27 µm, 4.5 µm, and 1.4 µm, respectively.

For these optimized base thicknesses, the current densities of the top GaInP subcell, middle GaAs subcell, and bottom InGaAs subcell are 12.30 mA/cm^2, 12.44 mA/cm^2, and 13.51 mA/cm^2, respectively (2D simulation mode), under illumination of AM1.5D spectrum. The subcells are current matched to within 10% at J_{sc}.

Simulating Subcell and Triple-Junction J–V Curves

The J–V curves of all the stacked subcells are calculated in the Sentaurus Workbench project `05-MJSC-Subcell-JV`; whereas, the J–V curve of the triple-junction solar cell is calculated in the `06-MJSC-JV` project. These projects also calculate the photovoltaic parameters (defined in Inspect: Light J–V Curves on page 23) of the stacked subcells and the triple-junction solar cell, respectively.

Figure 9 shows the light J–V curves of all the stacked subcells, and Figure 10 shows the light J–V and P–V curves of the triple-junction solar cell.

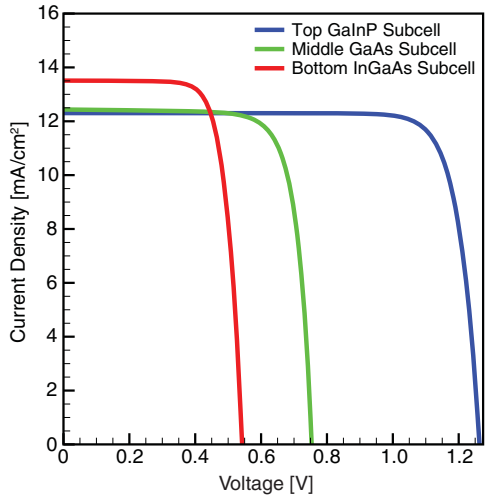

Figure 9 Light J–V characteristics of stacked subcells under AM1.5D illumination (2D simulation mode)

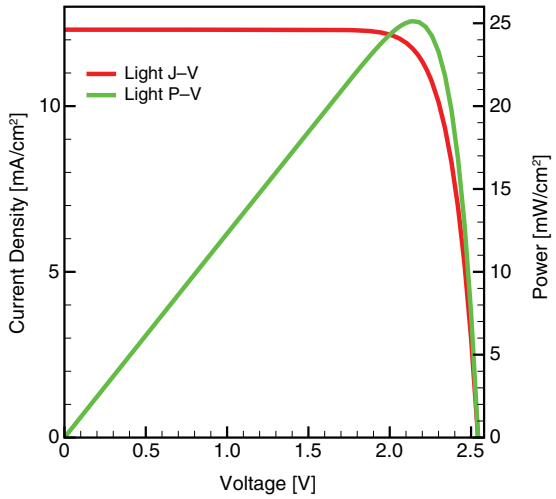

Figure 10 Light J–V and P–V characteristics of triple-junction solar cell under AM1.5D illumination (2D simulation mode)

In the 2D simulation mode, the triple-junction solar cell has a J_{sc} of 12.31 mA/cm^2, an V_{oc} of 2.55 V, a fill factor of 81%, and an efficiency of 25.5% under illumination of AM1.5D spectrum.

Simulating Triple-Junction EQE and Reflectance Spectra

The reflectance coefficients of the triple-junction solar cell and external quantum efficiency (EQE) spectra of the stacked subcells are simulated and plotted in the Sentaurus Workbench project `07-MJSC-Subcell-EQE-Reflect`.

Figure 11 shows the reflectance spectra as well as that the thickness of the individual layers in the double-layer MgF_2/TiO_x ARC has been designed such that the reflectance is negligible over a broad range of wavelengths.

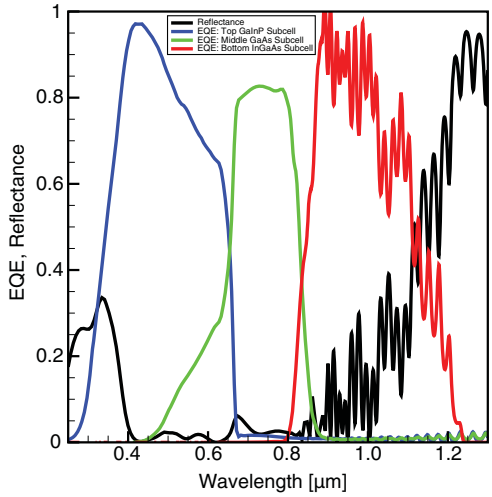

Figure 11 EQE spectra of subcells and reflectance spectrum of GaInP/GaAs/InGaAs triple-junction solar cell (2D simulation mode)

Figure 11 also shows the EQE spectra of the stacked subcells (Eq. 7 is used to calculate the EQE spectra). It also shows that the EQE of each subcell dominates in a particular wavelength region. The EQE of the GaInP subcell dominates in the lower wavelength (higher energy) region and is zero for higher wavelengths. Similarly, the EQE of the InGaAs subcell dominates in the higher wavelength (lower energy) region and is zero for lower wavelengths. This is due to the fact that each subcell absorbs photons in a different wavelength range. It can be seen that the EQE of all three subcells is greater than 80%.

General Simulation Setup: All Projects

This template project is organized into seven Sentaurus Workbench projects within the directory `MJ-SolarCell`:

- **01-Eqbm-Banddgm**: This project simulates the equilibrium energy band diagram (dark) of a hetero-structure, a triple-junction solar cell, or individual

single-junction solar cells. See Equilibrium Energy Band Diagram: 01-Eqbm-Banddgm on page 15.

- **02-TunnelDiode-JV**: This project simulates the J–V curve of the tunnel diode used in the triple-junction solar cell and plots its equilibrium energy band diagram. See Simulating Tunnel Diode: 02-TunnelDiode-JV on page 17.
- **03-SingleJn-JV-QE**: This project simulates the light J–V curve, QE spectra, and SR of the individual single-junction GaInP, GaAs, and InGaAs solar-cell structures and calculates their photovoltaic parameters. See Simulating Single-Junction Solar Cell: 03-SingleJn-JV-QE on page 19.
- **04-MJSC-Subcell-Balance-Jsc**: This project simulates the J_{sc} for each subcell in the triple-junction solar cell. It is used to optimize the base thickness of each subcell so that the optical absorption is equal and the subcells are current balanced at J_{sc}.
- **05-MJSC-Subcell-JV** and **07-MJSC-Subcell-EQE-Reflect**: The project `05-MJSC-Subcell-JV` simulates the light J–V curve, and the project `07-MJSC-Subcell-EQE-Reflect` simulates the EQE spectra of each subcell in the triple-junction solar cell.

  The project `07-MJSC-Subcell-EQE-Reflect` simulates the reflectance spectrum of the triple-junction solar cell and is used to design an ARC for the solar cell.
- **06-MJSC-JV**: This project simulates the light J–V and P–V curves of the triple-junction solar cell and calculates its photovoltaic parameters.

Common Simulation Setups

The following projects are called multijunction solar cell (MJSC) Sentaurus Workbench projects:

- `04-MJSC-Subcell-Balance-Jsc`
- `05-MJSC-Subcell-JV`
- `06-MJSC-JV`
- `07-MJSC-Subcell-EQE-Reflect`

The project `03-SingleJn-JV-QE` and all the MJSC projects are referred to as *solar-cell Sentaurus Workbench projects*.

In this section, the simulation setup common to all Sentaurus Workbench projects is discussed. For the simulation setup specific to a Sentaurus Workbench project, refer to the simulation setup section for that project.

The directory `MJ-SolarCell` was created using the Epi/MatPar template directory (`epi_template`) and the `epi` Sentaurus Workbench project, using the method discussed in the *Epi/MatPar User Guide*, Chapter 5 on page 85 [1].

For details about the directory structure and all folders and files in `epi_template`, refer to the *Epi/MatPar User Guide*, Table 1 on page 1 and Table 2 on page 2 [1] and to the application note *Template for Creating and Simulating Multilayered Heterostructure Devices with TCAD Sentaurus* [8].

The directory `MJ-SolarCell` also contains the subdirectory `spectra`, which comprises AM1.5D, AM1.5G, and AM0 solar spectra files.

The tool flow of all the Sentaurus Workbench projects is:

- Epi, which creates the multilayer stack (heterostructure, tunnel diode, or solar cell).
- Sentaurus Structure Editor, which creates the complete device (heterostructure, tunnel diode, or solar cell).
- MatPar, which creates the main parameter file and user-defined material parameter files from user-specified models.
- Sentaurus Device, which calculates the electrical and optical device characteristics.
- Inspect and Tecplot SV, which are the visualization tools.

The tool input and output files for Epi and MatPar are described in the *Epi/MatPar User Guide*, Table 2 on page 2 [1].

The files needed to set up the multilayer stack and to create the user-defined parameter files are:

- `epi_epi.csv`: Epi command file, which is a CSV file containing all relevant data for each layer of the multilayer stack.
- `MatPar_mpr.cmd`: MatPar command file, which is used to define some variables and invoke some procedures.
- `par/<material>.tcl` or `par/<material>.par`: Source parameter file, which MatPar uses to create the user-defined parameter files. `<material>.tcl` is used to specify the model names.
- `pardb/../<model>.tcl`: Model files. Material parameter calculations are implemented in the model files.

Epi

Epi creates the epitaxial multilayer stack (heterostructure, tunnel diode, or solar cell) by generating a Scheme script file that contains commands to generate the multilayer stack. This script file is sourced in the subsequent Sentaurus Structure Editor instance.

The Epi node in all the Sentaurus Workbench projects consists of the following Sentaurus Workbench parameter:

- `wtot` [µm]: Controls the width of the multilayer stack. It is set to `2` for `01-Eqbm-Banddgm` and `02-TunnelDiode-JV`. It is set to `500` for all the solar-cell projects.

In addition, it is possible to mesh the multilayer stack and store it in a TDR file by specifying the predefined Epi variable `generate` in the Global section of the CSV file. This structure can be used to verify the multilayer stack.

Sentaurus Structure Editor

Sentaurus Structure Editor creates the complete device by modifying the multilayer stack created by Epi and adding electrical contacts. The modifications are specific to the Sentaurus Workbench project. For example, for the MJSC projects, Sentaurus Structure Editor etches the cap layer, deposits the ARC, adds electrical contacts, and performs mesh refinement in the lateral (x) direction.

For all the Sentaurus Workbench projects, Sentaurus Structure Editor adds the electrical contacts – anode and cathode – to the structure. The location and width of these contacts is specific to the project.

After the structure is created, the generated mesh and doping information are stored in a TDR file, which is passed to Sentaurus Device.

MatPar

MatPar creates the main parameter file required by Sentaurus Device and user-defined (regionwise and materialwise) material parameter files from user-specified models. For each Sentaurus Workbench project, MatPar creates a separate set of material parameter files. For example, the MatPar node in the Sentaurus Workbench project `epi` creates:

- A materialwise user-defined material parameter file for GaAs.
- Regionwise user-defined material parameter files for the front surface field (fsf) and back surface field (bsf) regions of the multilayer stack.
- A materialwise interface parameter file for all GaAs–AlGaAs interfaces.

The MatPar node in all the Sentaurus Workbench projects consists of the following Sentaurus Workbench parameters:

- `temp` [K]: Operating temperature of the device. It is set to `300`.
- `tauSRH` [s]: Maximum GaAs minority carrier lifetime in the Shockley–Read–Hall (SRH) recombination model. It is set to `1e-9`.
- `srv` [cm/s]: Surface recombination velocity for all GaAs–AlGaAs interfaces. It is set to `1e2`.
- `model`: Name of the model used to compute some model parameters and to print the corresponding parameter sections. It is set to `Levinshtein`.

Tecplot SV

A Tecplot SV node is present in all Sentaurus Workbench projects except `epi` and `07-MJSC-Subcell-EQE-Reflect`. The Tecplot SV node plots the energy band diagram of the structure under equilibrium or under illumination at J_{sc}, with all the electrical contacts at zero bias. It consists of one Sentaurus Workbench parameter:

- `ycut`: Controls the x-axis value at which Tecplot SV creates a 1D cut perpendicular to the x-axis and draws the energy band diagram.

For all the Sentaurus Workbench projects, `ycut` is set to `wtot/2`. It is set to `1` for `01-Eqbm-Banddgm` and `02-TunnelDiode-JV`. It is set to `250` for all the solar-cell projects.

The diffusion lengths of electrons and holes for all the layers in the structure are also computed in the Tecplot SV node.

Tool-specific Setups: All Projects

As discussed in General Simulation Setup: All Projects on page 7, the simulations are organized into seven Sentaurus Workbench projects within the directory `MJ-SolarCell`. In this section, the tool-specific setups common to all simulation projects are discussed.

For the tool-specific setups of a particular Sentaurus Workbench project, refer to the relevant section for that project.

Generating Multilayer Stack Using Epi, Sentaurus Structure Editor, and Sentaurus Mesh

The complete device structure is created using Epi and Sentaurus Structure Editor instances in the Sentaurus Workbench project. Epi creates the multilayer stack using internal calls to Sentaurus Structure Editor and Sentaurus meshing tools, which are not explicitly included in Sentaurus Workbench. The subsequent instance of Sentaurus Structure Editor creates the complete device.

For example, the multilayer stack created by Epi (see Figure 12 on page 10 and Table 2 on page 34) in the Sentaurus Workbench project `epi` consists of several epitaxial layers. It has a GaAs p-n junction with an fsf layer

on top of the p-type emitter and a bsf layer below the n-type base. Both the fsf and bsf layers are AlGaAs layers. The bottom buffer layer is a GaAs layer. All layers have uniform doping concentration.

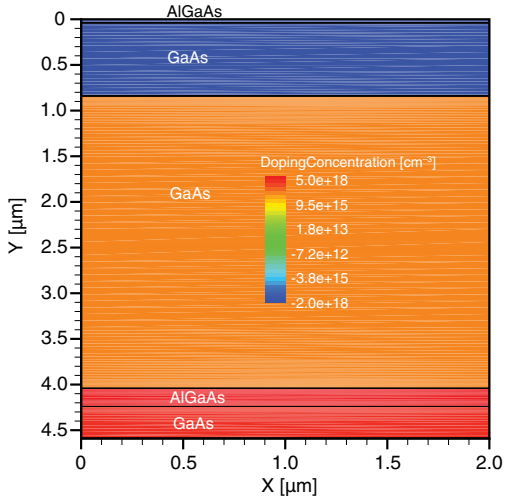

Figure 12 Multilayer stack generated in the epi Sentaurus Workbench project

The Epi command file `epi_epi.csv` contains all of the information needed to create the multilayer stack. It can be edited easily in either a text editor or a spreadsheet application, and it must be saved in the CSV format. For an example on how to save a file in CSV format in OpenOffice, refer to the *Epi/MatPar User Guide*, Saving command file in CSV format in OpenOffice on page 9 [1].

The file `epi_epi.csv` consists of three sections:

- Global
- Layers
- Material and Interface

The Global section is used to define Epi variables using the `global` command. The value of these variables is related to the complete multilayer stack. All Epi variables are available to Sentaurus Structure Editor in the subsequent node as Scheme variables (through `nX_epi.scm`, where `X` is the node number[1]) as well as to MatPar as Tcl variables (through `nX_epi.tcl`).

All statements in the Global section begin with one of the following two characters:

- The hash (#) is used to insert comments in the CSV file or to add Sentaurus Workbench preprocessor #-commands.
- The dollar sign ($) precedes the `global` command.

1. nX is part of the file name of the output files of both Epi and MatPar. In general, it is part of the file name of the output files of any TCAD Sentaurus tool.

There are two types of Epi variable: predefined and user-defined. For a complete list of predefined Epi variables, refer to the *Epi/MatPar User Guide*, Table 4 on page 11 [1]. The `global` command is used either to modify the predefined Epi variables or to create user-defined Epi variables using the syntax:

```
$global variable1=<value>, variable2=<value>, ...
```

The dimensions of the multilayer stack are specified using the predefined Epi variables in the Global section of the command file `epi_epi.csv`. For all the projects, Epi creates a 2D multilayer stack since `Zmax=Zmin=0` by default. The width of this stack, `Xmax-Xmin`, is specified using the Sentaurus Workbench variable `wtot` as follows:

```
$global Xmin=0, Xmax=@wtot@
```

The Sentaurus Workbench variable `wtot` is converted to a user-defined Epi variable:

```
$global wtot=@wtot@
```

`wtot` is used to insert a mesh line perpendicular to the x-direction at the center of the structure (`wtot/2`) in the Epi node of the projects `01-Eqbm-Banddgm` and `02-TunnelDiode-JV` and all the solar-cell projects (except `06-MJSC-JV`) in 1D simulation mode:

```
#if [string compare @dimension@ "1d"] == 0
$global dXmin=@<wtot/2>@, dXmax=@<wtot/2>@
#endif
```

The Sentaurus Workbench parameter `dimension` is defined in Epi on page 19. For the 1D simulation mode, it is set to `1`. Tecplot SV plots the energy band diagram of the multilayer stack along this mesh line.

Epi converts the `Epi` variable `wtot` to a Scheme variable `wtot`. For all the Sentaurus Workbench projects, the Scheme variable `wtot` is used in the subsequent Sentaurus Structure Editor node to place some of the electrical contacts.

Since the predefined Epi variable `generate` is set to `tdr`, Epi generates the multilayer stack through an internal call to Sentaurus Structure Editor and Sentaurus Mesh, and stores the structure in a TDR file (`nX_epi_msh.tdr`). This TDR file can be used to verify the multilayer stack by visualizing it in Tecplot SV.

Table 1 on page 33 shows the Layers section and Material and Interface section in tabular format. The Layers section in the CSV file consists of one statement for each layer to be created, starting with the top layer of the structure.

The layer description in each statement consists of several comma-separated fields:

```
<Region>, <Material>, <SourceParFile>, <Thickness>,
<Doping>, <MoleFraction>, <Refinement>
```

The fields contain information (region name, material, thickness, doping concentration, and mole fraction) about each layer. The mesh refinement strategy in the y-direction for each layer is also specified. Epi uses this information to create the multilayer stack.

The fields are described in the *Epi/MatPar User Guide*, Table 5 on page 13, and the command syntax for each field is described in Tables 5, 7, 8, and 9 [1].

The CSV file is also used to specify the following MatPar-specific information:

- The location and file names of the regionwise user-defined parameter files to be created by MatPar, by defining the predefined Epi variable `parFileMask` in the Global section:

```
$global parFileMask=./npar/
  n(node)_(material)_(xMole).par
```

- The name of the source parameter file in the `SourceParFile` field in the Layers section to be used by MatPar to create regionwise, materialwise, and materialwise interface user-defined parameter files.

For details about `parFileMask` and the procedure for creating various user-defined parameter files, refer to the *Epi/MatPar User Guide*, Material and Interface section on page 19 [1].

There are two types of source parameter file:

- `<material>.tcl` (also called the Tcl source parameter file)
- `<material>.par`

There is one source parameter file for each material or region. The Tcl source parameter files contain Tcl commands, and MatPar processes these files to create the user-defined parameter files. The `<material>.par` files have the same format as the Sentaurus Device parameter files. MatPar includes these files without processing in the user-defined material parameter files. For details, refer to the *Epi/MatPar User Guide*, Source parameter files on page 52 [1].

MatPar creates a regionwise user-defined parameter file if the name of the source parameter file is specified in the `SourceParFile` field of the corresponding layer definition in the Layers section. For example, for the Sentaurus Workbench project `epi`, the Tcl source parameter file `AlGaAs.tcl` is specified in the fsf and bsf layer definitions:

```
# Layers section
fsf,AlGaAs,AlGaAs.tcl,0.04,-2.00e18,0.8,(yref 0.01)
bsf,AlGaAs,AlGaAs.tcl,0.2,5.00e18,0.2,
  (mbox 0.01 1.2 both)
```

As a result, MatPar creates two regionwise user-defined parameter files, `nX_AlGaAs_0.8.par` and `nX_AlGaAs_0.2.par`, for the fsf and bsf AlGaAs regions, respectively.

MatPar creates a materialwise user-defined parameter file `n(node)_<material>.par` if:

- The `SourceParFile` field is empty in the Layers section for the regions for which a single materialwise parameter file is required.
- A statement for each such material is appended in the Material and Interface section.

For example, for all GaAs regions in the multilayer stack in the `epi` Sentaurus Workbench project, a single materialwise parameter file `nX_GaAs.par` is created using the Tcl source parameter file `GaAs.tcl` as a result of the following listing:

```
# Layers section
emitter,GaAs,,0.8,-9.00e17,,(mbox 0.05 1.1 both)
base,GaAs,,3.2,1.00e17,,(mbox 0.01 1.1 both)
buffer,GaAs,,0.35,2.00e18,,(mbox 0.01 1.2 both)
# Material and Interface section
,GaAs,GaAs.tcl,,1e18,,
```

To create materialwise parameter files for materials *not* included in the Epi-generated multilayer stack, the material name and the name of the source parameter file are specified in the Material and Interface section.

For example, the triple-junction solar-cell structure consists of a double-layer ARC, comprising MgF_2 and TiO_x. These layers are deposited in the Sentaurus Structure Editor node. To create materialwise parameter files for MgF_2 and TiO_x, the source parameter files `MgF.par` and `TiOx.par` are specified as follows:

```
# Material and Interface section
,MgF,MgF.par,,,,
,TiOx,TiOx.par,,,,
```

As a result, MatPar creates the user-defined parameter files `nX_MgF.par` and `TiOx.par`.

MatPar can create user-defined materialwise interface parameter files. For example, to create such a file for all the GaAs–AlGaAs interfaces in the multilayer stack, the following syntax is used in the Epi command file:

```
# Material and Interface section
,GaAs/AlGaAs,GaAs_AlGaAs.par,,,,
```

Here, all fields except `Material` and `SourceParFile` are empty. MatPar uses the source parameter file `GaAs_AlGaAs.par` to create the materialwise interface parameter file `nX_GaAs_AlGaAs.par`.

Device Generation Using Sentaurus Structure Editor and Sentaurus Mesh

The Scheme script file generated by Epi, `nX_epi.scm`, is sourced in the command file `sde_dvs.cmd` of the subsequent Sentaurus Structure Editor node:

```
(load "@pwd@/@episcm@")
```

where `@episcm@` refers to `nX_epi.scm`[1]. This file contains commands for creating the multilayer stack and mesh refinement in the x-direction. In addition, all of the Epi variables are defined as Scheme variables in this file.

The Sentaurus Structure Editor node in all the Sentaurus Workbench projects adds at least two electrodes, anode and cathode, to the multilayer stack created by Epi. The width and location of the electrodes depends on the specific Sentaurus Workbench project.

Many of the structure-modifying commands in Sentaurus Structure Editor are executed using the Scheme variables `Ytop`, `Ybot`, `Y0_<region>`, and `Y1_<region>`.

`Ytop` and `Ybot` are the y-coordinates of the top edge of the topmost layer and the bottom edge of the bottommost layer of the multilayer stack, respectively. `Y0_<region>` and `Y1_<region>` are the y-coordinates of the top and the bottom edges of a layer, respectively. Epi creates these variables.

For example, in the `epi` Sentaurus Workbench project, the cathode is placed at the bottom edge of the buffer layer using `Ybot` in the command:

```
(sdegeo:define-2d-contact (find-edge-id
  (position (/ wtot 2) Ybot 0)) "cathode")
```

Similarly, the anode is placed at the top edge of the fsf layer using `Ytop`:

```
(sdegeo:define-2d-contact (find-edge-id
  (position (/ wtot 2) Ytop 0)) "anode")
```

Here, the anode can also be placed at the top edge of the fsf layer using `Y0_fsf`:

```
(sdegeo:define-2d-contact (find-edge-id
  (position (/ wtot 2) Y0_fsf 0)) "anode")
```

In `epi`, the width of both anode and cathode is equal to `wtot`.

Sentaurus Structure Editor calls the meshing engine Sentaurus Mesh to generate the structure files for Sentaurus Device. This procedure generates a device structure written in a TDR file, containing doping and grid data.

1. In the *Epi/MatPar User Guide*, Table 3 on page 6 lists all file references defined in `epi/gtooldb.tcl`.

Generating Parameter File Using MatPar

MatPar is used to implement functional dependencies of material parameters as a function of doping, temperature, or mole fraction (or other physical variables), which are different from any of the Sentaurus Device built-in models. To obtain this flexibility through simple Tcl scripting, all such parameters must be available during the preprocessing stage. In addition, the parameters are assumed to be constant throughout a given layer.

For material parameters that depend on spatially varying structure parameters (such as the doping or mole-fraction profiles), or even solution variables (such as the self-consistent lattice temperature), the MatPar approach cannot be used. In such cases, use the Sentaurus Device mole-fraction and doping dependency parameter file (refer to the *Epi/MatPar User Guide*, Run-time and preprocessed models on page 67 [1]).

A parameter file consists of several parameter sections. Each parameter section corresponds to a particular material property and a default model. Each parameter section consists of values of several model parameters. The set of model parameters of a particular parameter section is called the *default model parameter set* of the parameter section.

For the purpose of creating user-defined parameter files using MatPar, the computation of the default model parameter set for each parameter section is implemented in separate files called *model files*. The model files reside in the Material Parameter Database.

For each parameter section to be printed in a user-defined parameter file, the model names are specified in a Tcl source parameter file.

For a particular material or region, MatPar uses these model names to compute the default model parameter set for each section and to print all of the sections in a regionwise or materialwise user-defined material parameter file.

The MatPar node has three types of input file:

- The MatPar command file `MatPar_mpr.cmd` is used to specify Tcl variables and to convert Sentaurus Workbench variables to Tcl variables. It also invokes some Tcl procedures.
- The source parameter files (`<material>.tcl` or `<material>.par`) are used to create the user-defined parameter files. The Tcl source parameter file `<material>.tcl` is used to specify:
  - The name of the model file for each parameter section.
  - The name of the procedure used to print each parameter section in the user-defined parameter file.

- The model files <model>.tcl are used to compute the default model parameter set for each parameter section. These are located in a three-level directory structure inside the directory pardb.

For details about the directory structure and model files, refer to the *Epi/MatPar User Guide*, Material Parameter Database on page 41 and Model files on page 58 [1].

The MatPar command file MatPar_mpr.cmd is a Tcl script file comprising some mandatory commands and some optional commands. These commands define a few variables and invoke some procedures. The complete list of variables is described in the *Epi/MatPar User Guide*, Table 14 on page 46 [1].

The Tcl command set defines the following Tcl variables in the MatPar command file:

- temp [K]: Operating temperature of the solar cell:
  - Set to @temp@, that is, to the temperature defined as the global Sentaurus Workbench parameter.
  - Used by MatPar to compute the default model parameter values for all parameter sections.
- substrate: Reference material for electron affinity:
  - Set to GaAs.
  - Used by MatPar to compute the electron affinity value for AlInP, GaInP, AlGaAs, and InGaAs.
- pardb: Location of the Material Parameter Database:
  - Set to "../pardb". The path of pardb is specified relative to the Sentaurus Workbench project.
  - All model files are located in MJ-SolarCell/pardb in a three-level directory structure [1].

In addition to defining these variables, nX_epi.tcl, which contains all of the information about the layered structure, is also sourced in the MatPar command file:

```
source "@pwd@/@epitcl@"
```

With this file, MatPar knows which source parameter file to process and the names of the corresponding user-defined parameter files. In addition, all variables defined in Epi as well as MatPar can be accessed from both the Tcl source parameter files and model files.

For each Sentaurus Workbench project, Epi uses the same set of source parameter files to create a different set of user-defined material parameter files for each Sentaurus Workbench project. For details, refer to the *Generating Parameter File Using MatPar* section of the specific project.

For example, in the epi Sentaurus Workbench project, as previously discussed, MatPar uses:

- GaAs.tcl to create the materialwise user-defined parameter files for GaAs, nX_GaAs.par.
- AlGaAs.tcl to create the following regionwise user-defined parameter files for the fsf and bsf regions, respectively:
  - nX_AlGaAs_0.8.par.
  - nX_AlGaAs_0.2.par.
- GaAs_AlGaAs.par to create a materialwise interface parameter file, nX_GaAs_AlGaAs.par, for all the GaAs–AlGaAs interfaces.

The source parameter files are located in MJ-SolarCell/par and are common to all Sentaurus Workbench projects in the MJ-SolarCell directory. The user-defined parameter files, which are created by MatPar, are stored in the npar subdirectory of each Sentaurus Workbench project; whereas, the main parameter file nX_mpr.par is stored in each Sentaurus Workbench project directory.

MatPar uses the source parameter files <material>.par to create user-defined parameter files by copying these source parameter files. For example, the MgF.par and TiOx.par source parameter files contain two parameter sections: Epsilon and TableODB. MatPar uses these source parameter files to create the user-defined parameter files nX_MgF.par and nX_TiOx.par.

All Tcl source parameter files can be accessed directly from Sentaurus Workbench. For details, refer to the *Epi/MatPar User Guide*, Source parameter files on page 52 [1].

Each Tcl source parameter file contains commands in a single Tcl command block. It consists of four sections:

- The Variable Definition section defines the variables, material, and conduction-band offset (cbo[1]).
- The Input Validation and Header Printing section validates the value of variables and prints the header section in user-defined parameter files.
- The Parameter Initialization section calculates the model parameter set value for each parameter section.
- The Parameter Printing section prints the parameter section in user-defined parameter files.

For details about these sections, refer to the *Epi/MatPar User Guide*, Source parameter files on page 52 [1]. You can easily create customized <material>.tcl files by using the template Tcl source parameter files (refer to the *Epi/MatPar User Guide*, Template Tcl source parameter file on page 58 [1]).

For example, the Variable Definition section of the Tcl source parameter file GaAs.tcl defines the material as GaAs:

```
set material GaAs
```

1. For a definition of cbo, refer to the *Epi/MatPar User Guide*, Simulating band structure of heterostructures on page 63 [1].

For all the Sentaurus Workbench projects, the reference material for the electron affinity calculation is defined as GaAs in the MatPar command file. Therefore, the Variable Definition section of all the Tcl source parameter files for all the other materials defines the valence-band discontinuity (VBO) value. VBO = ΔE_V, where ΔE_V is the difference in the valence-band edges of the material with respect to the reference material.

For example, for the GaAs–AlGaAs interface, `AlGaAs.tcl` defines the VBO value for AlGaAs with respect to GaAs:

```
set material AlGaAs
set VBO [expr {-$::xMole*0.46}]
```

Here, ΔE_V and, therefore, the VBO is the difference in the valence-band edges of GaAs and AlGaAs. It is negative since the valence-band edge of AlGaAs is lower than that of GaAs.

Similarly, for the GaAs–InGaAs interface, `InGaAs.tcl` defines the VBO value for InGaAs with respect to GaAs:

```
set material InGaAs
set VBO [expr {0.4*(1.247*0.3+1.5*(1.0-$::xMole)-
0.4*(1.0-$::xMole)*(1.0-$::xMole))}]
```

In this case, VBO is positive since the valence-band edge of InGaAs is higher than that of GaAs.

All the Tcl source parameter files contain a pair of `initProperty` and `print` statements for each parameter section to be printed in the user-defined parameter file. The model names for a particular parameter section are specified in the `initProperty` statement. For example, MatPar executes the following code in `GaAs.tcl` and computes the default model parameter set for the `Epsilon` section from the permittivity model from Piprek [9]:

```
initProperty $material Permittivity Epsilon Piprek
```

MatPar uses these computed default model parameter sets to print the `Epsilon` section in `nX_GaAs.par` after executing the following command:

```
${material}::Epsilon::print
```

MatPar can parameterize the default model parameters as well as the model names. For example, the Sentaurus Workbench parameter `model` is set to `Levinshtein` in all the Sentaurus Workbench projects. This model name is used in the Tcl source parameter files to print several parameter sections. As a result of the following statements in `GaAs.tcl`, MatPar computes the default model parameter set for the Levinshtein electron density-of-states mass model [10] and prints the `eDOSMass` section in the materialwise parameter file for GaAs (`nX_GaAs.par`):

```
initProperty $material BandStructure eDOSMass @model@
${material}::eDOSMass::print
```

The Tcl variable `tauSRH` is passed to the default SRH recombination model for GaAs through the following Tcl command in the file `pardb/GaAs/Recombination/Scharfetter/Default.tcl`:

```
variable taumax_n $::tauSRH
variable taumax_p $::tauSRH
```

This approach can be used to calibrate the default model parameter values for any parameter section.

This application provides two examples of how to customize the model files in the parameter database `pardb`:

- Specifying the VBO value: The parameter database `pardb` uses the cbo values to compute the electron affinity values of a material with respect to the reference material [1]. In this application, the VBO value is used instead of the cbo value. Therefore, the copy of `pardb` in the `MJ-SolarCell` directory contains an appropriately modified model file for the Piprek bandgap model [9] for AlInP, GaInP, AlGaAs, GaAs, and InGaAs.
- Modifying the default model file for the GaAs SRH recombination model: The copy of `pardb` in the `MJ-SolarCell` directory contains a modified version of the GaAs default SRH recombination model. Instead of specifying the value of `taumax_n` and `taumax_p` in the default model file, the Tcl variable `tauSRH` is used to specify these values.

Device Simulation Using Sentaurus Device

In the Sentaurus Workbench project `01-Eqbm-Banddgm`, Sentaurus Device simulates the equilibrium energy band diagram of a heterostructure, a triple-junction solar cell, or a single-junction solar cell. In `02-TunnelDiode-JV`, Sentaurus Device simulates the equilibrium energy band diagram as well as current–voltage characteristics of a tunnel diode.

For all the solar-cell projects, Sentaurus Device simulates electrical and optical characteristics of a solar cell (single-junction or triple-junction). Sentaurus Device calculates the optical generation rate in solar cells and couples it with the electrical simulation. Several models are implemented in Sentaurus Device, enabling the simulation of optical carrier generation. Refer to the *Sentaurus Device User Guide*, in particular, the chapter of optical generation for detailed information about the photogeneration models and associated keywords.

The TMM is used to find the rate of optical carrier generation for different wavelengths of the incident radiation, which is used subsequently in Inspect to calculate the photogenerated current and to deduce the efficiency of the device.

To compute the current density in units of mA/cm^2, an `AreaFactor` is specified in the `Physics` section of the Sentaurus Device command file for all Sentaurus Workbench projects:

```
Physics {
  AreaFactor = @< 1e11/wtot >@
}
```

Tecplot SV

The Tecplot SV node plots the energy band diagram of the device under equilibrium or at J_{sc} by using a Tecplot SV macro and a TDR file. For the equilibrium energy band diagram simulation, the TDR file stores the solution of the Poisson equation. For the energy band diagram at J_{sc} simulation, the solution of the Poisson equation, and electron and hole continuity equations is stored. The file name of the TDR file is specific to the project.

The orthogonal slicer in Tecplot SV creates a 1D cut perpendicular to the x-axis at `x=ycut` and draws the energy band diagram. For all the Sentaurus Workbench projects, `ycut` is set to `wtot/2`. Therefore, the energy band diagram of the device is plotted along the mesh line at `wtot/2` in the projects `01-Eqbm-Banddgm` and `02-TunnelDiode-JV`, and in all the solar-cell projects (except `06-MJSC-JV`) in the 1D simulation mode.

Equilibrium Energy Band Diagram: 01-Eqbm-Banddgm

Project Setup

This project simulates the equilibrium energy band diagram (dark) of an AlInP/GaInP/AlGaAs/GaAs/InGaAs heterostructure, a triple-junction solar cell, or individual single-junction solar cells. The simulation of the equilibrium energy band diagram of the heterostructure is used to verify the band structure alignment in the heterostructure.

Epi

As discussed in Epi on page 8, Epi creates a multilayer stack with a width, `wtot`. Here, the Sentaurus Workbench parameter `wtot` is set to `2` µm.

In addition to `wtot`, the Epi node consists of the following Sentaurus Workbench parameter:

- `structure`: Set to `hetero|multijn|GaInP|GaAs|InGaAs`. Controls the type of multilayer stack that Epi creates.

If `structure=hetero`, an AlInP/GaInP/AlGaAs/GaAs/InGaAs heterostructure is created. The heterostructure consists of a single layer of all the different materials used in the triple-junction solar cell. It is used to verify the band offsets.

If `structure=multijn`, a GaInP/GaAs/InGaAs triple-junction solar cell is created.

Setting `structure=GaInP|GaAs|InGaAs` creates the individual single-junction GaInP, GaAs, or InGaAs solar-cell structures, respectively.

Moreover, Epi consists of the following Sentaurus Workbench parameters:

- `dtopbase` [µm]: Set to `0.27`. Controls the thickness of the base layer in the GaInP single-junction solar cell or the GaInP subcell in the triple-junction solar cell.
- `dmidbase` [µm]: Set to `4.5`. Controls the thickness of the base layer in the GaAs single-junction solar cell or GaAs subcell in the triple-junction solar cell.
- `dbotbase` [µm]: Set to `1.4`. Controls the thickness of the base layer in the InGaAs single-junction solar cell or InGaAs subcell in the triple-junction solar cell.

Sentaurus Structure Editor

As discussed in Sentaurus Structure Editor on page 9, Sentaurus Structure Editor modifies the multilayer stack created by Epi and converts it into a device by adding the electrical contacts: anode and cathode.

Sentaurus Structure Editor adds the cathode and anode at the top and bottom of the structure, respectively. The width of both the anode and cathode is equal to `wtot`.

MatPar

MatPar creates the main parameter file, and user-defined regionwise and materialwise material parameter files. The MatPar node consists of the Sentaurus Workbench parameters `temp`, `tauSRH`, `srv`, and `model`. See MatPar on page 9.

The user-defined parameter files created by MatPar depend on the Sentaurus Workbench parameter `structure` as follows:

- If `structure=hetero`, MatPar creates a regionwise user-defined parameter file for each layer.
- If `structure=GaInP|GaAs|InGaAs`, MatPar creates a materialwise user-defined parameter file for the emitter and base regions for all cases. For each single-junction solar-cell structure, the fsf and bsf regions consist of the same ternary semiconductor with the same mole fraction. For the GaInP and GaAs subcells, MatPar creates a single materialwise parameter file for these regions. For InGaAs subcells, MatPar creates a single regionwise parameter file for these regions. For

the GaInP and InGaAs subcells, MatPar also creates a materialwise user-defined parameter file for GaAs since GaAs is the reference material for electron affinity.

- If `structure=multijn`, MatPar creates:
  - Materialwise user-defined parameter files for all the $Al_{0.5}In_{0.5}P$, $Ga_{0.5}In_{0.5}P$, $Al_{0.3}Ga_{0.7}As$, GaAs, and $In_{0.3}Ga_{0.7}As$ regions.
  - A materialwise interface parameter file for all the GaAs–AlGaAs interfaces.

Sentaurus Device

Sentaurus Device solves the Poisson equation with both electrodes at 0 V, and the result is used to plot the energy band diagram of the multilayer stack.

Tecplot SV

The Tecplot SV node plots the equilibrium energy band diagram of the multilayer stack at `ycut=1` (see Tecplot SV on page 9).

Tool-specific Setups

Generating Multilayer Stack Using Epi, Sentaurus Structure Editor, and Sentaurus Mesh

In this project, the Epi node creates a multilayer stack based on the value of the Sentaurus Workbench parameter `structure`. The complete structure of the GaInP/GaAs/InGaAs triple-junction solar cell is specified in Table 1 on page 33. Epi creates either the complete triple-junction solar cell multilayer stack or only a part of the complete structure. For example, if `structure=GaInP`, only the top GaInP subcell from Table 1 is created. All the layers have uniform doping concentration.

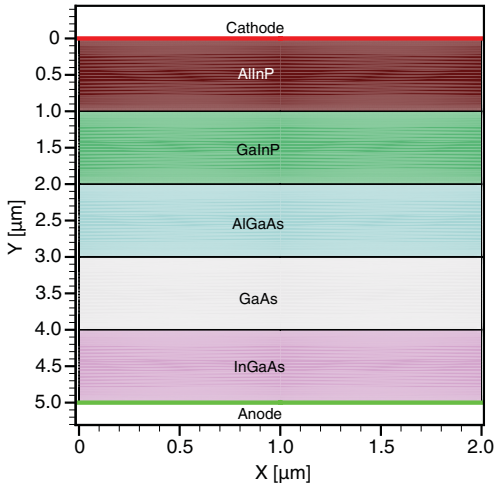

Figure 13 AlInP/GaInP/AlGaAs/GaAs/InGaAs heterostructure

In addition, Epi can create the AlInP/GaInP/AlGaAs/GaAs/InGaAs heterostructure shown in Figure 13. This heterostructure is described in Table 3 on page 34. The heterostructure consists of a single layer of all the different materials used in the triple-junction solar-cell structure.

For details, see Generating Multilayer Stack Using Epi, Sentaurus Structure Editor, and Sentaurus Mesh on page 9.

Device Generation Using Sentaurus Structure Editor and Sentaurus Mesh

Sentaurus Structure Editor adds the electrical contacts, cathode and anode at the top and bottom of the multilayer stack using the Scheme variables `Ytop` and `Ybot`, respectively. The width of both the anode and cathode is equal to the width of the structure, `wtot` (see Device Generation Using Sentaurus Structure Editor and Sentaurus Mesh on page 12).

Generating Parameter File Using MatPar

The user-defined parameter files created by MatPar depend on the Sentaurus Workbench parameter `structure` as follows:

- If `structure=hetero`, MatPar uses `AlInP.tcl`, `GaInP.tcl`, `AlGaAs.tcl`, `GaAs.tcl`, and `InGaAs.tcl` to create the region-wise parameter files, `nX_AlInP_0.5.par`, `nX_GaInP_0.5.par`, `nX_AlGaAs_0.3.par`, `nX_GaAs.par`, and `nX_InGaAs_0.7.par`, respectively.
- If `structure=GaInP`, MatPar uses:
  - `AlInP.tcl` to create a materialwise user-defined parameter file `nX_AlInP_0.5.tcl` for the topfsf and topbsf regions.
  - `GaInP.tcl` to create a materialwise user-defined parameter file `nX_GaInP_0.5.tcl` for the topem and topbase regions.
  - `GaAs.tcl` to create a materialwise user-defined parameter file `nX_GaAs.par` for GaAs.
- If `structure=GaAs`, MatPar uses:
  - `GaInP.tcl` to create a materialwise user-defined parameter file `nX_GaInP_0.5.tcl` for the midfsf and midbsf regions.
  - `GaAs.tcl` to create a materialwise user-defined parameter file `nX_GaAs.tcl` for the midem and midbase regions.
- If `structure=InGaAs`, MatPar uses:
  - `GaInP.tcl` to create a regionwise user-defined parameter file `nX_GaInP_0.25.tcl` for the botfsf and botbsf regions.
  - `InGaAs.tcl` to create a materialwise user-defined parameter file `nX_InGaAs_0.7.tcl` for the botem and botbase regions.

- `GaAs.tcl` to create a materialwise user-defined parameter file `nX_GaAs.par` for GaAs.

- If `structure=multijn`, MatPar uses:
  - `AlInP.tcl`, `GaInP.tcl`, `AlGaAs.tcl`, `GaAs.tcl`, and `InGaAs.tcl` to create materialwise user-defined parameter files for all the $Al_{0.5}In_{0.5}P$, $Ga_{0.5}In_{0.5}P$, $Al_{0.3}Ga_{0.7}As$, GaAs, and $In_{0.3}Ga_{0.7}As$ regions, respectively.
    As a result, the materialwise parameter files `nX_AlInP_0.5.par`, `nX_GaInP_0.5.par`, `nX_AlGaAs_0.3.par`, `nX_GaAs.par`, and `nX_InGaAs_0.7.par` are created.
  - `GaInP.tcl` to create a regionwise user-defined parameter file `nX_GaInP_0.25.tcl` for the botfsf and botbsf regions.
  - `GaAs_AlGaAs.par` to create a materialwise interface parameter file `nX_GaAs_AlGaAs.par` for all GaAs–AlGaAs interfaces.

See Generating Parameter File Using MatPar on page 12.

Device Simulation Using Sentaurus Device

As discussed in Device Simulation Using Sentaurus Device on page 14, the equilibrium energy band diagram of the multilayer stack (heterostructure, triple-junction solar cell, or single-junction solar cell) is simulated in the Sentaurus Device node.

Sentaurus Device solves the Poisson equation with both anode and cathode at 0 V, and the solution is stored in the file `nX_Eqbm_Banddgm_des.tdr`:

```
Solve {
  Poisson
  Plot (FilePrefix = "n@node@_Eqbm_Banddgm")
}
```

Tecplot SV uses the file `nX_Eqbm_Banddgm_des.tdr` to visualize the equilibrium energy band diagram of the multilayer stack.

Since the purpose of this project is to verify the VBO values of various materials with respect to GaAs, the bandgap narrowing model is not activated if `structure=hetero`.

Tecplot SV

The Tecplot SV node plots the equilibrium energy band diagram of the multilayer stack (heterostructure, triple-junction solar cell, or single-junction solar cell) using the TDR file `nX_Eqbm_Banddgm_des.tdr` (see Tecplot SV on page 15).

Figure 4 on page 5 shows the equilibrium energy band diagram of the AlInP/GaInP/GaAs/InGaAs heterostructure.

Simulating Tunnel Diode: 02-TunnelDiode-JV

Project Setup

This project simulates the J–V curve of the tunnel diode used in the triple-junction solar cell and plots its equilibrium energy band diagram. It can be used to calibrate the nonlocal tunneling parameters [3] and to design the tunnel diode so that the peak tunneling current density in the tunnel diode is greater than the J_{sc} of the triple-junction solar cell [6].

Epi

Epi creates a multilayer stack with width `wtot`. In this node, Epi creates the tunnel-diode multilayer stack with a width `wtot` of 2 µm . See Epi on page 8.

Sentaurus Structure Editor

Sentaurus Structure Editor modifies the multilayer stack created by Epi and converts it into a device by adding the electrical contacts, anode and cathode (see Sentaurus Structure Editor on page 9).

Sentaurus Structure Editor adds the anode and cathode to the top and bottom of the structure, respectively. The width of both the anode and cathode is equal to `wtot`.

MatPar

MatPar creates the main parameter file and user-defined regionwise and materialwise material parameter files (see MatPar on page 9). The MatPar node consists of the Sentaurus Workbench parameters `temp`, `tauSRH`, `srv`, and `model`.

MatPar creates separate regionwise parameter files for the AlGaAs and GaAs layers, and a materialwise interface parameter file for the GaAs–AlGaAs interface.

Sentaurus Device

Sentaurus Device performs voltage sweeps and calculates the current. The solution of the Poisson equation with both electrodes at 0 V is used to plot the equilibrium energy band diagram of the tunnel diode.

Inspect

Inspect plots the J–V characteristics of the tunnel diode and computes the peak tunneling current density (J_{peak}) in mA/cm^2 and the corresponding peak voltage (V_{peak}) in volts.

Tecplot SV

The Tecplot SV node plots the equilibrium energy band diagram of the tunnel diode at `ycut=1` (see Tecplot SV on page 9).

Tool-specific Setups

Generating Multilayer Stack Using Epi, Sentaurus Structure Editor, and Sentaurus Mesh

The triple-junction solar-cell structure (described in Table 1 on page 33) consists of two p^+-AlGaAs/n^+-GaAs tunnel diodes. The tunnel diodes serve as interconnects between the stacked subcells.

Both tunnel diodes have the same structure, and each tunnel diode consists of a heavily doped p-type AlGaAs top layer and a heavily doped n-type GaAs bottom layer. Both the layers have uniform doping concentration. Both the tunnel diodes have the same thickness and doping concentration.

In this project, the Epi node creates a multilayer stack comprising a single tunnel-diode structure (see Figure 14). The AlGaAs region is called the *phighTD region* and the GaAs region is called the *nhighTD region*.

For details, see Generating Multilayer Stack Using Epi, Sentaurus Structure Editor, and Sentaurus Mesh on page 9.

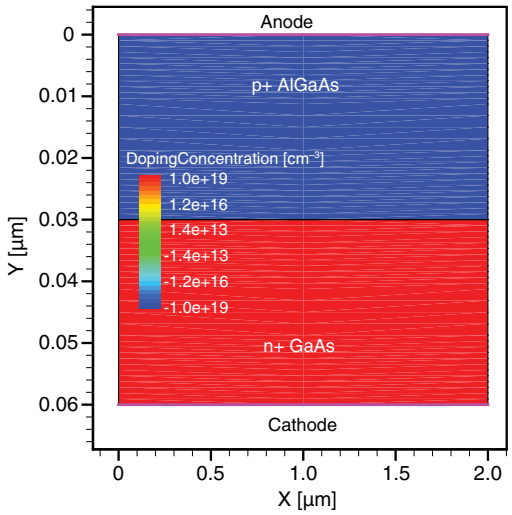

Figure 14 AlGaAs/GaAs tunnel-diode structure

Device Generation Using Sentaurus Structure Editor and Sentaurus Mesh

In this project, Sentaurus Structure Editor adds the electrical contacts, anode and cathode, at the top and bottom of the tunnel diode using the Scheme variables `Ytop` and `Ybot`, respectively.

The width of both the anode and cathode is equal to the width of the structure, `wtot` (see Device Generation Using Sentaurus Structure Editor and Sentaurus Mesh on page 12).

Generating Parameter File Using MatPar

MatPar uses:

- `AlGaAs.tcl` to create a regionwise user-defined parameter file `nX_AlGaAs_0.3.par` for the phighTD region.
- `GaAs.tcl` to create a regionwise user-defined parameter file `nX_GaAs.par` for the nhighTD region.
- `GaAs_AlGaAs.par` to create a materialwise interface parameter file `nX_GaAs_AlGaAs.par` for the GaAs–AlGaAs interface.

See Generating Parameter File Using MatPar on page 12.

Sentaurus Device: Equilibrium Energy Band Diagram

The equilibrium energy band diagram of the tunnel diode is simulated in this Sentaurus Device node. Sentaurus Device solves the Poisson equation with both the anode and cathode at 0 V, and the solution is stored in the file `nX_Eqbm_Banddgm_des.tdr`:

```
Solve {
  Poisson
  Plot (FilePrefix = "n@node@_Eqbm_Banddgm")
}
```

The file `nX_Eqbm_Banddgm_des.tdr` is used by Tecplot SV to visualize the equilibrium energy band diagram of the tunnel diode.

See Device Simulation Using Sentaurus Device on page 14.

Sentaurus Device: J–V Simulation

Sentaurus Device simulates the J–V characteristics of the tunnel diode using the nonlocal tunneling model. Refer to the *Sentaurus Device User Guide*, in particular, the chapter on tunneling for details about the nonlocal tunneling model and associated keywords.

The tunneling parameters g_C and g_V are specified in the `BarrierTunneling` section of the material parameter file `GaAs_AlGaAs.par`. The tunneling masses m_C and m_V for GaAs and AlGaAs are specified in the `BarrierTunneling` section of the parameter files `GaAs.tcl` and `AlGaAs.tcl`, respectively.

Sentaurus Device simulates the tunneling current at the phighTD–nhighTD interface.

A special-purpose ‘nonlocal’ mesh named TD_NLM is constructed for this interface using the following specification in the global Math section:

```
NonLocal "TD_NLM" (
  RegionInterface = "phighTD/nhighTD"
  Length = 15e-7
  Permeation = 15e-7
)
```

Length determines the maximum tunneling distance, and Permeation determines the length by which Sentaurus Device extends the nonlocal lines, across the phighTD–nhighTD interface, towards the opposite side for which the line is constructed.

The nonlocal tunneling model is activated for the nonlocal mesh TD_NLM for both electrons and holes in the global Physics section:

```
eBarrierTunneling "TD_NLM" (
  Band2Band
  TwoBand
)
hBarrierTunneling "TD_NLM" (
  Band2Band
  TwoBand
)
```

Sentaurus Device simulates the J–V characteristics by ramping the anode voltage using the Quasistationary statement in the Solve section.

Tecplot SV

The Tecplot SV node plots the equilibrium energy band diagram of the AlGaAs/GaAs tunnel diode using the TDR file nX_Eqbm_Banddgm_des.tdr (see Figure 5 on page 5). For details, see Tecplot SV on page 15.

Inspect

The forward-biased J–V curve of the AlGaAs/GaAs tunnel diode is plotted in this node (see Figure 6 on page 6). The data for obtaining the J–V curve is contained in the standard plot files. The J_{peak} and the corresponding peak voltage (V_{peak}) are calculated using Inspect commands [11].

Simulating Single-Junction Solar Cell: 03-SingleJn-JV-QE

Project Setup

This project simulates the light J–V curve, QE spectra, and SR of the individual single-junction GaInP, GaAs, and InGaAs solar-cell structures. It also calculates the photovoltaic parameters of these solar cells. This project is used to verify that the individual subcell structures work properly. It is also used to optimize the y-direction mesh in the subcells and, therefore, the y-direction mesh in the triple-junction solar cell.

Epi

Epi creates a multilayer stack with width, wtot. In this node, Epi creates a GaInP, a GaAs, or an InGaAs single-junction solar-cell multilayer stack with width wtot of 500 µm (see Epi on page 8).

In addition to wtot, the Epi node consists of the following Sentaurus Workbench parameters:

- dimension: Set to 1d|2d. Controls the simulation mode of the structure. If dimension=1d, 1D simulation mode is used. If dimension=2d, 2D simulation is used.
- subcell: Set to GaInP|GaAs|InGaAs. Controls the type of single-junction solar cell created by Epi. Setting subcell=GaInP|GaAs|InGaAs creates a single-junction GaInP, GaAs, or InGaAs solar cell, respectively.
- dtopbase [µm]: Set to 0.27. Controls the thickness of the base layer in the GaInP single-junction solar cell.
- dmidbase [µm]: Set to 4.5. Controls the thickness of the base layer in the GaAs single-junction solar cell.
- dbotbase [µm]: Set to 1.4. Controls the thickness of the base layer in the InGaAs single-junction solar cell.

Sentaurus Structure Editor

Sentaurus Structure Editor modifies the multilayer stack created by Epi and converts it into a device (see Sentaurus Structure Editor on page 9).

It creates a single-junction solar cell by adding electrical contacts, cathode and anode, at the top and bottom of the structure, respectively.

If dimension=2d, mesh refinements in the lateral (x) direction are performed. The Sentaurus Structure Editor node consists of one Sentaurus Workbench parameter:

- wfrontc [µm]: Set to 5. If dimension=2d, it controls the half-width of the front contact finger. If dimension=1d, wfrontc has no significance.

The width of the anode is equal to wtot. The width of the cathode is equal to wtot if dimension=1d or to wfrontc if dimension=2d.

MatPar

MatPar creates the main parameter file, and user-defined regionwise and materialwise material parameter files. The MatPar node consists of the Sentaurus Workbench

parameters `temp`, `tauSRH`, `srv`, and `model` (see MatPar on page 9).

The user-defined parameter files created by MatPar depends on the Sentaurus Workbench parameter `subcell`.

If `subcell=GaInP|GaAs|InGaAs`, MatPar creates a materialwise user-defined parameter file for the emitter and base regions for all cases. For each single-junction solar cell, the fsf and bsf regions consist of the same ternary compound semiconductor with the same mole fraction. For the GaInP and GaAs subcells, MatPar creates a single materialwise parameter file for these regions. For the InGaAs subcells, MatPar creates a single regionwise parameter file for these regions. For the GaInP and InGaAs subcells, MatPar also creates a materialwise user-defined parameter file for GaAs since GaAs is the reference material for electron affinity.

Sentaurus Device: JV Instance

Sentaurus Device performs voltage sweeps and calculates the current in the solar cell under illumination. It consists of the following Sentaurus Workbench parameter:

- `spectrum`: Set to `am1.5g|am1.5d|am0`. Controls the solar spectrum used to illuminate the solar cell. Here, it is set to `am1.5d`.

TMM is used to calculate the optical generation rate in the solar cell under illumination.

The solution of the Poisson equation, and the electron and hole continuity equations with both electrodes at 0 V is used to simulate the energy band diagram of the solar cell at J_{sc}.

Sentaurus Device: QE Instance

This instance of Sentaurus Device consists of the following Sentaurus Workbench parameters:

- `wstart` [µm]: Initial wavelength; set to `0.3`.
- `wend` [µm]: Final wavelength; set to `1.1` for the GaInP and GaAs solar cells; and set to `1.3` for the InGaAs solar cell.
- `wsteps`: Number of steps between the initial and final wavelengths; set to `60`.
- `intensity` [W m^{-2}]: Incident light intensity; set to `1`.

Sentaurus Device simulates the QE spectra and SR by ramping the wavelength of incident light from `wstart` to `wend`, in steps of `wsteps`.

Inspect: Plot_JV Instance

This Inspect node plots the illuminated J–V and P–V characteristics of the single-junction solar cell. It also extracts the photovoltaic parameters, V_{oc}, J_{sc}, fill factor (FF), and efficiency (η).

Inspect: Plot_QE Instance

This Inspect node plots the QE spectra and the SR (internal and external) of the single-junction solar cell.

Tecplot SV

The Tecplot SV node plots the energy band diagram of the single-junction solar cell at J_{sc} at `ycut=250` (see Tecplot SV on page 9).

Tool-specific Setups

Generating Multilayer Stack Using Epi, Sentaurus Structure Editor, and Sentaurus Mesh

The triple-junction solar cell structure (described in Table 1 on page 33) consists of three subcells stacked on top of each other. Each subcell consists of:

- An n-type emitter on top of a p-type base.
- An n-type fsf layer on top of the emitter.
- A p-type bsf layer at the bottom of the base.

In all the subcells, the emitter and base layers are the same material and, for the GaInP and InGaAs cells, they have the same mole fraction. Similarly, the fsf and bsf layers consist of the same material and mole fraction. All layers have uniform doping concentration. Epi creates one of these subcells based on the Sentaurus Workbench parameter `subcell`. For example, if `subcell=InGaAs`, the bottom InGaAs subcell structure of Table 1 is created by Epi. The structure created by Epi is the same for both the 1D and 2D simulation mode.

For details, see Generating Multilayer Stack Using Epi, Sentaurus Structure Editor, and Sentaurus Mesh on page 9.

Device Generation Using Sentaurus Structure Editor and Sentaurus Mesh

Sentaurus Structure Editor adds the electrical contacts, cathode and anode, at the top and bottom of the subcell using the Scheme variables `Ytop` and `Ybot`, respectively.

The single-junction subcell structure can be simulated in either the 1D or 2D simulation mode by controlling the Sentaurus Workbench variable `dimension`.

If `dimension=1d`, Sentaurus Structure Editor creates the single-junction solar-cell structure in the 1D simulation

mode. If `dimension=2d`, Sentaurus Structure Editor creates the structure in the 2D simulation mode. Both structures are shown in Figure 15 for a single-junction InGaAs solar cell.

The multilayer stack created by Epi is the same in both the 1D and 2D simulation mode. The difference is in the size of the electrodes. In the 2D simulation mode, the electrodes are placed such that the smallest symmetry element of the single-junction subcell is simulated.

In both 1D and 2D simulation mode, the width of the anode equals `wtot`. The width of the cathode depends on the simulation mode: It equals `wtot` in the 1D simulation mode and `wfrontc` in the 2D simulation mode, as shown in Figure 15.

For details, see Device Generation Using Sentaurus Structure Editor and Sentaurus Mesh on page 12.

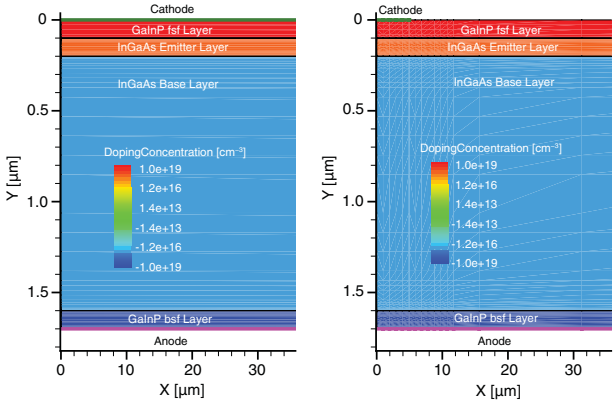

Figure 15 InGaAs single-junction solar-cell structure created by Sentaurus Structure Editor in (*left*) 1D simulation mode and (*right*) 2D simulation mode

Generating Parameter File Using MatPar

The user-defined parameter files created by MatPar depend on the Sentaurus Workbench parameter `subcell` as follows:

- If `subcell=GaInP`, MatPar uses:
  - `AlInP.tcl` to create a materialwise user-defined parameter file `nX_AlInP_0.5.tcl` for the topfsf and topbsf regions.
  - `GaInP.tcl` to create a materialwise user-defined parameter file `nX_GaInP_0.5.tcl` for the topem and topbase regions.
  - `GaAs.tcl` to create a materialwise user-defined parameter file `nX_GaAs.par` for GaAs.
- If `subcell=GaAs`, MatPar uses:
  - `GaInP.tcl` to create a materialwise user-defined parameter file `nX_GaInP_0.5.tcl` for the midfsf and midbsf regions.
  - `GaAs.tcl` to create a materialwise user-defined parameter file `nX_GaAs.tcl` for the midem and midbase regions.
- If `subcell=InGaAs`, MatPar uses:
  - `GaInP.tcl` to create a regionwise user-defined parameter file `nX_GaInP_0.25.tcl` for the botfsf and botbsf regions.
  - `InGaAs.tcl` to create a materialwise user-defined parameter file `nX_InGaAs_0.7.tcl` for the botem and botbase regions.
  - `GaAs.tcl` to create a materialwise user-defined parameter file `nX_GaAs.par` for GaAs.

For details, see Generating Parameter File Using MatPar on page 12.

Sentaurus Device: Equilibrium Energy Band Diagram

The equilibrium energy band diagram of a single-junction solar cell is simulated in the project `01-Eqbm-Banddgm` by setting the Sentaurus Workbench parameter `structure` to `GaInP|GaAs|InGaAs` (see Device Simulation Using Sentaurus Device on page 17).

Sentaurus Device: Energy Band Diagram at Jsc

The `JV` instance of Sentaurus Device simulates the energy band diagram at J_{SC} of a single-junction solar cell under illumination of AM1.5D solar radiation.

Sentaurus Device uses TMM to compute the optical generation profile of the single-junction solar cell.

The TMM is initialized by the keyword `TMM` in the `Physics` section:

```
OpticalGeneration (
  QuantumYield = 1
  ComputeFromSpectrum
)
Optics (
  TMM (
    NodesPerWavelength = 100
    Excitation (
      Theta = 0
      Polarization = 0.5
    )
    Stripe (
#if [string compare @dimension@ "2d"] == 0
      Left = @wfrontc@
#elseif [string compare @dimension@ "1d"] == 0
      Left = 0
#endif
      Right = @wtot@
    )
  )
)
```

Currently, TMM supports the optical field of normal incidence (`Theta=0`) with equal TE and TM components of the polarization.

QuantumYield controls the number of electron–hole pairs generated per photon.

The keyword ComputeFromSpectrum specifies that the incident light spectrum is given in the external file as specified in the File section of the sdevice_des.cmd file:

```
IlluminationSpectrum = "../spectra/am15dWm2-
short.txt"
```

When using TMM, the optical field is calculated first on an intrinsic mesh, specified by the keyword NodesPerWavelength, which controls the number of points used in the calculation of the optical field.

The Stripe section specifies the left and right boundaries of the planar layered structure to which TMM is applied and determines the width over which the solar cell is illuminated. For the 1D simulation mode, the entire solar cell is illuminated; whereas, for the 2D simulation mode, only the region outside the front contact finger is illuminated.

Sentaurus Device solves the Poisson equation, and the electron and hole continuity equations with all the electrodes at 0 V, and the solution is saved in the TDR file nX_Banddgm_Jsc_des.tdr using the following commands in the Solve section:

```
Poisson
Coupled (Iterations = 100) { Poisson Electron Hole }
Plot (FilePrefix = "n@node@_Banddgm_Jsc")
```

Tecplot SV uses this TDR file to plot the energy band diagram of the triple-junction solar cell at J_{sc}.

Sentaurus Device: J–V Simulation

The J–V characteristics of the illuminated solar cell is calculated in the JV instance of Sentaurus Device. Similar to the simulation of the energy band diagram at J_{sc}, TMM is initialized by the keyword TMM in the Physics section (see Sentaurus Device: Energy Band Diagram at Jsc on page 21).

Sentaurus Device ramps the anode voltage to simulate the J–V characteristics of the single-junction solar cell. The voltage ramping in the Solve section for the solar cell under illumination is separated into two Quasistationary statements. The first one is with a larger voltage step:

```
#if "@subcell@" == "GaInP"
  Quasistationary (
    InitialStep = 5e-2
    MaxStep = 1e-1
    MinStep = 1e-5
    Goal{ voltage = 0.75 Name = "anode" }
    ){ Coupled {Poisson Electron Hole} }
#elif "@subcell@" == "GaAs"
  Quasistationary (
...
#elif "@subcell@" == "InGaAs"
  Quasistationary (
    InitialStep = 5e-2
    MaxStep = 1e-1
    MinStep = 1e-5
    Goal{ voltage = 0.2 Name = "anode" }
  ){ Coupled {Poisson Electron Hole} }
#endif
```

The goal bias voltage for the anode depends on the subcell being simulated.

The second Quasistationary statement is with a reduced voltage step to resolve the maximum power point of the solar cell:

```
Quasistationary (
  InitialStep = 5e-2
  MaxStep = 5e-2
  MinStep = 1e-5
  Goal{ voltage = 2 Name = "anode" }
){ Coupled {Poisson Electron Hole} }
```

Sentaurus Device: QE Spectra and Spectral Response

The quantum efficiency (QE) spectra of the illuminated single-junction solar cell is calculated in the QE instance of Sentaurus Device by activating TMM and ramping the wavelength. Similar to the simulation of energy band diagram at J_{sc}, TMM is initialized by the keyword TMM in the Physics section (see Sentaurus Device: Energy Band Diagram at Jsc on page 21).

The incident light wavelength and intensity are specified using the Sentaurus Workbench parameters with the keywords Wavelength and WavePower, respectively:

```
OpticalGeneration (
  QuantumYield = 1
)
Optics (
  TMM (
    NodesPerWavelength = 100
    Excitation (
      Theta = 0
      Polarization = 0.5
      Wavelength = @wstart@
      WavePower = @intensity@
    )
    Stripe (
#if [string compare @dimension@ "1d"] == 0
      Left = 0
#elseif [string compare @dimension@ "2d"] == 0
      Left = @wfrontc@
#endif
      Right = @wtot@
    )
  )
)
```

The Inspect node calculates the QE and spectral response (SR) from the integral of the optical generation rate as a function of wavelength. The `CurrentPlot` section is needed to plot output quantities such as the integral of the optical generation rate as a function of wavelength:

```
CurrentPlot {
  Model = "Optics"
  Parameter = "Wavelength"
  OpticalGeneration(Integrate(Semiconductor))
}
```

The integral of the optical generation rate is written in the standard `Plot` file when the wavelength ramp is performed in the `Quasistationary` statements in the `Solve` section. For the InGaAs single-junction solar cell, two `Quasistationary` statements are used:

```
Quasistationary (
  InitialStep = @<1./wsteps>@
  MaxStep = @<1./wsteps>@
  Minstep = 1e-5
  Goal { Model = "Optics" Parameter = "Wavelength"
         value = 0.65 }
  ) { Coupled {Poisson Electron Hole} }
Quasistationary (
  InitialStep = @<1./wsteps>@
  MaxStep = @<1./wsteps/5.>@
  Minstep = 1e-5
  Goal { Model = "Optics" Parameter = "Wavelength"
         value = @wend@ }
  ) { Coupled {Poisson Electron Hole} }
```

The second `Quasistationary` statement uses finer wavelength steps to resolve the oscillations in the EQE spectra and external spectral response (ESR).

Tecplot SV

The Tecplot SV node plots the energy band diagram of a single-junction solar cell at J_{sc} using the TDR file `nX_Banddgm_Jsc_des.tdr`. Figure 16 shows the energy band diagram of the InGaAs single-junction solar cell. For details, see Tecplot SV on page 15.

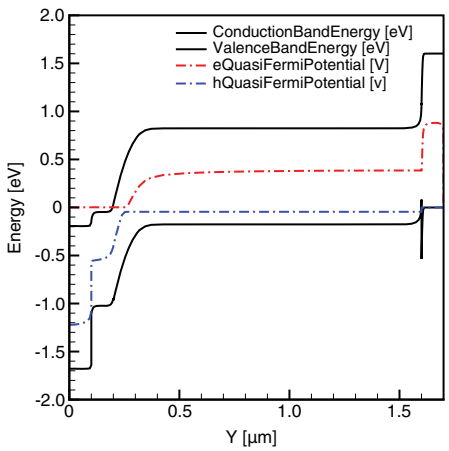

Figure 16 Energy band diagram of InGaAs single-junction solar cell

Inspect: Light J–V Curves

The light J–V curve and QE spectra are calculated in this node. The data for obtaining the light J–V curve is contained in the standard plot files. The power density of the illuminated solar cell [12]:

$$P = JV \tag{1}$$

is calculated using the Inspect command `cv_createWithFormula`.

Figure 17 shows the J–V and P–V characteristics of the illuminated InGaAs single-junction solar cell. The photocurrent is opposite to the dark current of the forward-biased diode. For a low applied voltage, the photocurrent is independent of the voltage and dominates the J–V curve of the illuminated device.

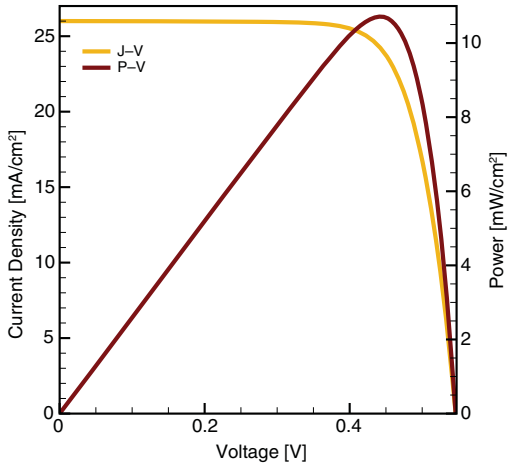

Figure 17 Light J–V and P–V characteristics of InGaAs single-junction solar cell

In this regime, the solar-cell power grows linearly and reaches a maximum at the operating point or maximum power point of the solar cell. When the voltage is increased after the maximum power point is reached, the contribution of the photocurrent to the total current through the diode becomes less important, and the total current changes sign. The power density at the maximum power point is P_m, and the corresponding voltage and current density are V_m and J_m.

The photovoltaic parameters, J_{sc}, V_{oc}, maximum power density (P_m), fill factor (FF), and efficiency (η) [12] are calculated using Inspect commands [11]. The fill factor is given by:

$$\mathrm{FF} = \frac{J_m \cdot V_m}{J_{sc} \cdot V_{oc}} = \frac{P_m}{J_{sc} \cdot V_{oc}} \tag{2}$$

The efficiency is given by:

$$\eta = \frac{P_m}{P_s} \tag{3}$$

where P_s is the power density of the incident light.

Inspect: QE Spectra

Inspect performs a simple calculation to obtain the internal and external quantum efficiencies and SR.

The internal quantum efficiency (IQE) is defined as the ratio of the number of carriers contributing to the electrical current under short-circuit conditions to the total number of photogenerated carriers. The latter can be found through the carrier generation rate integrated over the semiconductor volume. The IQE is given by:

$$\mathrm{IQE}(\lambda) = \frac{J_{sc}(\lambda)}{J_{ph}(\lambda)} \tag{4}$$

The external quantum efficiency (EQE) is defined as the ratio of the number of carriers contributing to the electrical current under short-circuit conditions to the number of incident photons of a particular energy:

$$\mathrm{EQE}(\lambda) = \frac{J_{sc}(\lambda)/q}{I_{in}/E_{ph}(\lambda)} \tag{5}$$

where I_{in} is the incident light intensity, and E_{ph} is the photon energy:

$$E_{ph} = \frac{hc}{\lambda} \tag{6}$$

Therefore:

$$\mathrm{EQE}(\lambda) = \frac{J_{sc}(\lambda)hc}{q\lambda I_{in}} \tag{7}$$

where h is Planck's constant, c is the speed of light, and λ is the wavelength of incident light.

The spectral responses, either internal (ISR) or external (ESR), are calculated from the corresponding quantum efficiency using [13]:

$$\mathrm{SR}(\lambda) = \frac{q\lambda}{hc}\mathrm{QE}(\lambda) \tag{8}$$

Using Eq. 4 to Eq. 8, the IQE and EQE spectra as well as the corresponding spectral response (ISR or ESR) are obtained using the Inspect command `cv_createWithFormula`. The results are shown in Figure 18 for the InGaAs single-junction solar cell.

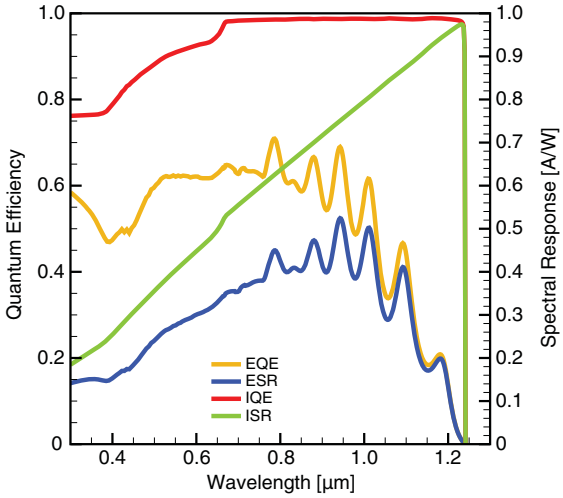

Figure 18 QE spectra and SR (internal and external) of InGaAs single-junction solar cell

At longer wavelengths, as the wavelength decreases, the internal spectral response (ISR) rises below the cut-off wavelength for InGaAs because InGaAs can absorb photons below the cut-off wavelength. As the wavelength decreases further, the ISR reduces because the InGaAs solar cell cannot use the energy of the absorbed photons to generate carriers [13].

Simulating Multijunction Solar Cells: MJSC Projects

Project Setup: All MJSC Projects

The MJSC Sentaurus Workbench projects simulate the electrical and optical characteristics of a triple-junction solar cell.

Epi

Epi creates a multilayer stack with a width, `wtot`. In this node, Epi creates a GaInP/GaAs/InGaAs triple-junction solar-cell stack with a width `wtot` of 500 µm (see Epi on page 8).

In addition to `wtot`, the Epi node consists of the following Sentaurus Workbench parameters:

- `dimension`: Set to `1d|2d`. Controls the simulation mode of the structure. If `dimension=1d`, the simulation is performed in 1D mode. If `dimension=2d`, the simulation is performed in 2D mode.
- `dtopbase [µm]`: Set to `0.27`. Controls the thickness of the base layer in the top GaInP subcell.

- `dmidbase [µm]`: Set to `4.5`. Controls the thickness of the base layer in the middle GaAs subcell.
- `dbotbase [µm]`: Set to `1.4`. Controls the thickness of the base layer in the bottom InGaAs subcell.

Sentaurus Structure Editor

Sentaurus Structure Editor modifies the multilayer stack created by Epi and converts it into a device (see Sentaurus Structure Editor on page 9).

Sentaurus Structure Editor modifies the triple-junction multilayer stack created by Epi and creates a triple-junction solar cell. It performs the following modifications:

- Adds the electrodes, cathode and anode, for all the MJSC Sentaurus Workbench projects.
- Adds the electrodes, toptjContact and bottjContact, for `04-MJSC-Subcell-Balance-Jsc`, `05-MJSC-Subcell-JV`, and `07-MJSC-Subcell-EQE-Reflect`.
- Etches the cap layer, deposits the ARC, and performs mesh refinement in the x-direction in the 2D simulation mode.

The cathode is placed at the top of the multilayer stack, and the anode is placed at the bottom. If `dimension=2d`, mesh refinements in the lateral (x) direction are performed.

The Sentaurus Structure Editor node consists of one Sentaurus Workbench parameter:

- `wfrontc [µm]`: Set to `5`. If `dimension=2d`, it controls the half-width of the front contact finger. If `dimension=1d`, `wfrontc` has no significance.

The width of the anode is equal to `wtot`. The width of the cathode is equal to `wtot` if `dimension=1d` and to `wfrontc` if `dimension=2d`.

If `dimension=2d`, the cap layer is etched from the regions outside the front contact finger and a double-layer ARC is deposited. If `dimension=1d`, the cap layer is not present in the structure and the ARC is deposited above the cathode.

For all the MJSC Sentaurus Workbench projects except `06-MJSC-JV`, a dummy triple-junction solar-cell structure is created by adding two extra contacts, toptjContact and bottjContact, with width equal to `wtot`. These are added at the p-n junction interface of the top and bottom tunnel diode, respectively.

MatPar

MatPar creates the main parameter file and user-defined regionwise and materialwise material parameter files. The MatPar node consists of the Sentaurus Workbench parameters `temp`, `tauSRH`, `srv`, and `model` (see MatPar on page 9).

MatPar creates these parameter files as follows:

- Materialwise user-defined parameter files for all the $Al_{0.5}In_{0.5}P$, $Ga_{0.5}In_{0.5}P$, $Al_{0.3}Ga_{0.7}As$, GaAs, and $In_{0.3}Ga_{0.7}As$ regions.
- A single regionwise user-defined parameter file for the botfsf and botbsf regions.
- Materialwise user-defined parameter files for MgF_2 and TiO_x layers for 2D simulation mode.
- Materialwise interface parameter file for all GaAs–AlGaAs interfaces.

Tecplot SV

The Tecplot SV node is present in all MJSC projects except `07-MJSC-Subcell-EQE-Reflect`.

The Tecplot SV node plots the energy band diagram of the triple-junction solar cell structure at J_{sc} at `ycut=250` (see Tecplot SV on page 9).

Project Setup: 04-MJSC-Subcell-Balance-Jsc, 05-MJSC-Subcell-JV, and 06-MJSC-JV

Sentaurus Device: Optics Instance

With all electrodes at 0 V, Sentaurus Device computes the optical generation profile of the triple-junction solar cell under illumination and stores the data in a TDR file. TMM is used to compute the optical generation rate in the solar cell under illumination.

The `Optics` instance of Sentaurus Device consists of the following Sentaurus Workbench parameter:

- `spectrum`: Set to `am1.5g|am1.5d|am0`. Controls the solar spectrum used to illuminate the solar cell. Here, it is set to `am1.5d`.

Sentaurus Device: sdevice Instance

This instance of Sentaurus Device loads the optical generation profile saved in the `Optics` instance of Sentaurus Device. The loaded optical generation profile is ramped by adding a time-transient Gaussian function to the profile. In the transient simulation, Sentaurus Device solves the Poisson equation, and the electron and hole continuity equations with all electrodes at 0 V, and the solution is used to:

- Calculate J_{sc} for each subcell in the project `04-MJSC-Subcell-Balance-Jsc`.

- Plot the energy band diagram of the triple-junction solar cell at J_{sc}.

For the `05-MJSC-Subcell-JV` and `06-MJSC-JV` projects, after the transient sweep, Sentaurus Device performs voltage sweeps and calculates the current.

Project Setup: 04-MJSC-Subcell-Balance-Jsc

This project simulates the J_{sc} for each subcell in the triple-junction solar cell. It can be used to design the base thickness of each subcell so that the optical absorption is equal and the cells are current balanced at J_{sc}.

Inspect

Inspect computes the J_{sc} for each subcell in the triple-junction solar cell. `Jsctop`, `Jscmid`, and `Jscbot` correspond to the top, middle, and bottom J_{sc} of the top, middle, and bottom subcell, respectively.

Project Setup: 05-MJSC-Subcell-JV

This project simulates the light J–V curve of each subcell in the triple-junction solar cell.

Sentaurus Device

This Sentaurus Device instance contains one Sentaurus Workbench parameter:

- `subcell`: Set to `GaInP|GaAs|InGaAs`. Controls the stacked subcell for which Sentaurus Device simulates the J–V curve.

Inspect

This Inspect node plots the illuminated J–V and P–V characteristics of all the stacked subcells of the triple-junction solar cell. It also extracts the photovoltaic parameters, V_{oc}, J_{sc}, fill factor (FF), and efficiency (η) for each subcell.

Project Setup: 06-MJSC-JV

This project simulates the light J–V curve of the triple-junction solar cell and calculates the photovoltaic parameters.

Inspect

Inspect plots the illuminated J–V and P–V characteristics, and extracts the photovoltaic parameters of the triple-junction solar cell.

Project Setup: 07-MJSC-Subcell-EQE-Reflect

This project simulates the EQE spectra of each stacked subcell in the triple-junction solar cell. It also simulates the reflectance spectrum of the triple-junction solar cell, which can be used to design an ARC for the triple-junction solar cell.

Sentaurus Device: eqe Instance

The `eqe` instance of Sentaurus Device simulates the EQE spectra of each subcell in the triple-junction solar cell. It consists of the following Sentaurus Workbench parameters:

- `wstart` [µm]: Initial wavelength; set to `0.3`.
- `wend` [µm]: Final wavelength; set to `1.3`.
- `wsteps`: Number of steps between the initial and final wavelengths; set to `100`.
- `intensity` [W m^{-2}]: Incident light intensity; set to `1`.

Sentaurus Device simulates the EQE spectra by ramping the wavelength of incident light from `wstart` to `wend`, in steps of `wsteps`.

Sentaurus Device: reflect Instance

The `reflect` instance of Sentaurus Device simulates the reflectance spectrum of the triple-junction solar cell. The simulation is performed by ramping the wavelength of incident light from `wstart` to `wend`, in steps of `wsteps`. The intensity of the incident light is equal to `intensity` for all wavelengths. The values of these Sentaurus Workbench parameters from the `eqe` instance are used (see Sentaurus Device: eqe Instance).

Inspect

Inspect node plots the EQE spectra of all the stacked subcells of the triple-junction solar cell. It also plots the reflectance spectrum of the triple-junction solar cell.

Tool-specific Setups

Generating Multilayer Stack Using Epi, Sentaurus Structure Editor, and Sentaurus Mesh

Solar cells are large-area semiconductor devices and have dimensions in the order of centimeters. Simulating the entire 3D structure involves huge computational resources. However, the structure of a GaInP/GaAs/InGaAs triple-junction solar cell is highly symmetric. Therefore, the simulation domain can be reduced to a 2D structure

consisting of a cut perpendicular to the front contact metallization grid. The smallest symmetry element spreads out from the middle of the front contact finger to the midpoint between the two front contact fingers [14]. The width of this symmetry element, `wtot`, is 500 µm in this Sentaurus Workbench project. Figure 1 on page 3 shows a portion of the smallest symmetry element for the GaInP/GaAs/InGaAs triple-junction solar cell.

The triple-junction solar-cell structure is created using Epi and Sentaurus Structure Editor instances in the Sentaurus Workbench project. Epi creates the multilayer stack using internal calls to Sentaurus Structure Editor and Sentaurus meshing tools, which are not explicitly included in Sentaurus Workbench. The subsequent instance of Sentaurus Structure Editor creates the complete solar-cell device.

The GaInP/GaAs/InGaAs triple-junction solar cell consists of several epitaxial layers [2]. It consists of three subcells GaInP, GaAs, and InGaAs, stacked on top of each other. The top GaInP subcell is series connected to the middle GaAs subcell, and the middle GaAs subcell is series connected to the bottom InGaAs subcell through two tunnel-diode interconnects.

Each subcell consists of:

- An n-type emitter on top of a p-type base.
- An n-type fsf layer on top of the emitter.
- A p-type bsf layer at the bottom of the base.

In all the subcells, the emitter and base layers are the same material and, for the GaInP and InGaAs cells, they have the same mole fraction. Similarly, the fsf and bsf layers consist of the same material and mole fraction. All layers have uniform doping concentration.

The triple-junction solar-cell multilayer stack is created by Epi and is shown in Figure 19. The details of the multilayer stack are specified in Table 1 on page 33.

The triple-junction solar-cell structure can be simulated in either 1D or 2D simulation mode by controlling the Sentaurus Workbench variable `dimension`. If `dimension=1d`, Epi creates the triple-junction multilayer stack in the 1D simulation mode, in which case, the GaAs cap layer is not present. If `dimension=2d`, Epi creates the triple-junction multilayer stack in the 2D simulation mode, as shown in Figure 19.

MatPar does not create any parameter file for the bottommost gold layer since a source parameter file is not specified in the Epi command file:

```
bcon,Gold,,0.1,,,(yref 0.05)
```

Sentaurus Device uses the default Sentaurus Device parameters for Gold.

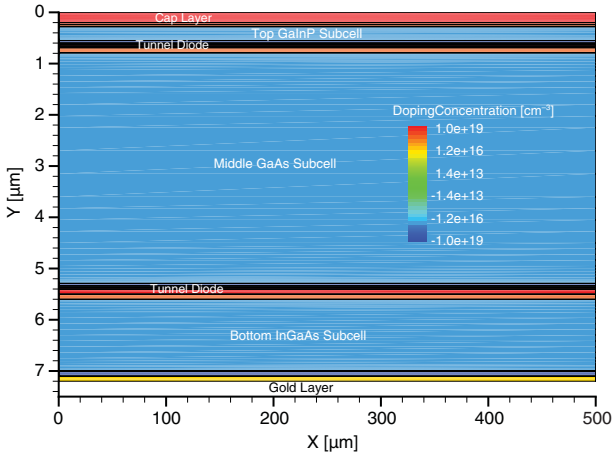

Figure 19 Triple-junction multilayer stack generated by Epi (2D simulation mode)

For details, see Generating Multilayer Stack Using Epi, Sentaurus Structure Editor, and Sentaurus Mesh on page 9.

Device Generation Using Sentaurus Structure Editor and Sentaurus Mesh

Sentaurus Structure Editor adds the electrical contacts, cathode and anode, at the top and bottom of the subcell using the Scheme variables `Ytop` and `Ybot`, respectively. In both 1D and 2D simulation mode, the width of the anode equals `wtot`, which is the width of the structure (see Device Generation Using Sentaurus Structure Editor and Sentaurus Mesh on page 12).

The width of the cathode depends on the simulation mode. It is equal to the width of the structure, `wtot`, for the 1D simulation mode. For the 2D simulation mode, it is equal to `wfrontc`:

```
#if [string compare @dimension@ "2d"] == 0
(sdegeo:define-2d-contact (find-edge-id
  (position (/ wfrontc 2) Ytop 0)) "cathode")
#elseif [string compare @dimension@ "2d"] == 0
(sdegeo:define-2d-contact (find-edge-id (position
  (/ wtot 2) Ytop 0)) "cathode")
#endif
```

For all the MJSC Sentaurus Workbench projects except `06-MJSC-JV`, a dummy triple-junction solar-cell structure is created by adding two extra contacts, toptjContact and bottjContact, with width equal to `wtot`. toptjContact and bottjContact are added at the p-n junction interface of the top and bottom tunnel diode, respectively.

For the 2D simulation mode, the cap-layer region outside the front contact finger is etched, and a double-layer ARC consisting of a top MgF_2 layer and a bottom TiO_x layer is deposited. This structure is generated in the Sentaurus Structure Editor node and is shown in Figure 1 on page 3.

For the 1D simulation mode, the ARC is deposited above the cathode. All these differences in the triple-junction solar-cell structures created by Sentaurus Structure Editor for 1D and 2D simulation modes are highlighted in Figure 20.

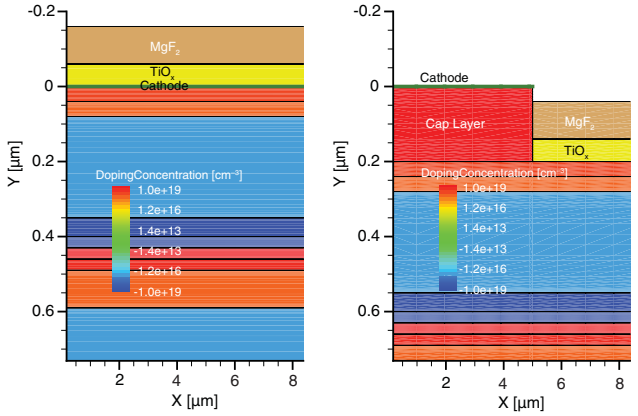

Figure 20 Part of triple-junction solar-cell structure created in (*left*) 1D simulation mode and (*right*) 2D simulation mode

The above steps are performed using the Scheme variables `Ytop`, `Y0_<region>`, and `Y1_<region>`. For example, in the 2D simulation mode, the cap layer is removed using `Y0_topfsf` in the commands:

```
#if [string compare @dimension@ "2d"] == 0
(define removeMe
  (sdegeo:create-rectangle (position wfrontc 0 0)
  (position wtot Y0_topfsf 0) Gold" "temp"))
(entity:delete removeMe)
#endif
```

In addition to the commands for modifying the multilayer stack, the Sentaurus Structure Editor node is used to place additional mesh refinement commands. In the 2D simulation mode, a finer mesh is placed under the front contact in the x-direction. Away from the contacts, the mesh becomes coarser.

MatPar

MatPar uses:

- `AlInP.tcl`, `GaInP.tcl`, `AlGaAs.tcl`, `GaAs.tcl`, and `InGaAs.tcl` to create materialwise user-defined parameter files for all the $Al_{0.5}In_{0.5}P$, $Ga_{0.5}In_{0.5}P$, $Al_{0.3}Ga_{0.7}As$, GaAs, and $In_{0.3}Ga_{0.7}As$ regions, respectively. As a result, the regionwise parameter files `nX_AlInP_0.5.par`, `nX_GaInP_0.5.par`, `nX_AlGaAs_0.3.par`, `nX_GaAs.par`, and `nX_InGaAs_0.7.par` are created.
- `GaInP.tcl` to create a regionwise user-defined parameter file `nX_GaInP_0.25.tcl` for the botfsf and botbsf regions.
- `MgF.par` and `TiOx.par` to create the materialwise user-defined parameter files `nX_MgF.par` and `nX_TiOx.par` for the MgF_2 and TiO_x regions, respectively. These files are created only in 2D simulation mode.
- `GaAs_AlGaAs.par` to create a materialwise interface parameter file `nX_GaAs_AlGaAs.par` for all GaAs–AlGaAs interfaces.

For details, see Generating Parameter File Using MatPar on page 12.

Sentaurus Device: Equilibrium Energy Band Diagram

The equilibrium energy band diagram of the triple-junction solar cell is simulated in the project `01-Eqbm-Banddgm` by setting the Sentaurus Workbench parameter `structure` to `multijn` (see Device Simulation Using Sentaurus Device on page 17).

Sentaurus Device: Energy Band Diagram of Dummy Triple-Junction Solar Cell at Jsc

The energy band diagram of the triple-junction solar cell at J_{sc} is simulated in the project `06-MJSC-JV`; whereas, the projects `04-MJSC-Subcell-Balance-Jsc` and `05-MJSC-Subcell-JV` simulate the energy band diagram of the dummy triple-junction solar cell at J_{sc}.

All these projects contain two Sentaurus Device nodes. The first one, `Optics`, uses TMM to compute the optical generation profile of the triple-junction solar cell under illumination of AM1.5D solar radiation. TMM is initialized by the keyword `TMM` in the `Physics` section:

```
OpticalGeneration (
  QuantumYield = 1
  ComputeFromSpectrum
)
Optics (
  TMM (
    NodesPerWavelength = 100
    Excitation (
      Theta = 0
      Polarization = 0.5
    )
    Stripe (
#if [string compare @dimension@ "2d"] == 0
      Left = @wfrontc@
#elseif [string compare @dimension@ "1d"] == 0
      Left = 0
#endif
      Right = @wtot@
    )
  )
)
```

Currently, the TMM supports the optical field of normal incidence (`Theta=0`) with equal TE and TM components of the polarization.

QuantumYield controls the number of electron–hole pairs generated per photon.

The keyword ComputeFromSpectrum specifies that the incident light spectrum is given in the external file as specified in the File section of the sdevice_des.cmd file:

```
IlluminationSpectrum = "../spectra/am15dWm2-
short.txt"
```

When using TMM, the optical field is calculated first on an intrinsic mesh, specified by the keyword NodesPerWavelength, which controls the number of points used in the calculation of the optical field.

The Stripe section specifies the left and right boundaries of the planar layered structure to which TMM is applied and determines the width over which the solar cell is illuminated. For a 1D simulation, the entire solar cell is illuminated; for a 2D simulation, only the region outside the front contact finger is illuminated.

Sentaurus Device calculates the optical generation profile using the statement:

```
Solve { Optics }
```

The optical generation profile is saved in a TDR file using the following statement in the File section:

```
OpticalGenerationOutput = "n@node@_OptGen_des.tdr"
```

This is loaded in the second instance of Sentaurus Device using the following statement in the File section:

```
OpticalGenerationFile = "n@previous@_OptGen_des.tdr"
```

The loaded optical generation profile is ramped by adding a time-transient Gaussian function to the profile using the following statements in the 04-MJSC-Subcell-Balance-Jsc and 05-MJSC-Subcell-JV projects:

```
Physics {
  OpticalGenerationFromFile (
    WaveTime = (0.9, 10)
    WaveTSigma = 0.3
    OptScaling = 1.0
  )
}
Solve {
  Transient (
    InitialStep = 1e-2
    MaxStep = 0.2
    MinStep = 1e-5
    Increment = 1.5
    InitialTime = 0
    FinalTime = 1
  )
{ Coupled (Iterations = 50) {Poisson Electron Hole} }
Plot (FilePrefix = "n@node@_Banddgm_Jsc")
}
```

In this transient simulation, Sentaurus Device solves the Poisson equation, and the electron and hole continuity equations with all electrodes at 0 V, and the solution is saved in the TDR file nX_Banddgm_Jsc_des.tdr. Tecplot SV uses this TDR file to plot the energy band diagram of the dummy triple-junction solar cell at J_{sc}.

The above time-transient ramping of the optical generation profile is equivalent to switching on the light slowly and aids in convergence of the initial solution of the Poisson equation, and the electron and hole continuity equations.

Sentaurus Device: Energy Band Diagram and J–V Simulation of Triple-Junction Solar Cell

The J–V characteristics of the illuminated triple-junction solar cell are calculated in the second Sentaurus Device node of 06-MJSC-JV by ramping the anode voltage. This simulation also involves the time-transient ramping of optical generation profile described in Sentaurus Device: Energy Band Diagram of Dummy Triple-Junction Solar Cell at Jsc on page 28.

After this step, the Solve section in this Sentaurus Device node contains two Quasistationary statements, one with the larger voltage step:

```
Quasistationary (
  InitialStep = 1e-1
  MaxStep = 2e-1
  MinStep = 1e-5
  Increment = 1.5
  DoZero
  Goal{ voltage = 1.7 Name = "anode" }
){ Coupled {Poisson Electron Hole}
Plot( FilePrefix = "n@node@_Banddgm_Jsc"
  Time = (0) )
}
```

and one with a reduced step to resolve the maximum power point of the triple-junction solar cell:

```
Quasistationary (
  InitialStep = 4e-2
  MaxStep = 4e-2
  MinStep = 1e-5
  Increment = 1.5
  Goal{ voltage = 2.7 Name = "anode" }
){ Coupled {Poisson Electron Hole} }
```

Here, the solution of the Poisson equation, and electron and hole continuity equations is stored in the file nX_Banddgm_Jsc_des.tdr at the beginning of the first Quasistationary statement. This file is used to plot the energy band diagram of the triple-junction solar cell in the Tecplot SV node.

Sentaurus Device: J–V Simulation of Stacked Subcells

The J–V characteristics of the illuminated subcells in the triple-junction solar cell are calculated in the second Sentaurus Device node of `05-MJSC-Subcell-JV` by ramping the bias of appropriate electrodes. This simulation also involves the time-transient ramping of optical generation profile described in Sentaurus Device: Energy Band Diagram of Dummy Triple-Junction Solar Cell at Jsc on page 28 and a single `Quasistationary` ramp.

The J–V curve of the GaInP subcell is simulated by ramping the cathode voltage; whereas, that of InGaAs subcell is simulated by ramping the anode voltage.

The J–V curve of the GaAs subcell is simulated by simultaneously ramping the bottjContact and anode voltage:

```
Quasistationary (
  InitialStep = 1e-2
  MaxStep = 1e-2
  MinStep = 1e-2
  Increment = 1.4
  DoZero
  Goal{ voltage = 1.0 Name = "bottjContact" }
  Goal{ voltage = 1.0 Name = "anode" }
){ Coupled {Poisson Electron Hole} }
```

Sentaurus Device: Simulating EQE Spectra

The EQE spectra of the subcells in the triple-junction solar cell is simulated in the `eqe` node of Sentaurus Device of the project `07-MJSC-Subcell-EQE-Reflect`.

This simulation involves wavelength ramping and the activation of TMM in the `Physics` section. The incident light wavelength is specified using the Sentaurus Workbench parameter `wstart` with the keyword `Wavelength`:

```
OpticalGeneration (
  QuantumYield = 1
)
Optics (
  TMM (
    NodesPerWavelength = 100
    Excitation (
      Theta = 0
      Polarization = 0.5
      Wavelength = @wstart@
      WavePower = 0
    )
    Stripe (
#if [string compare @dimension@ "2d"] == 0
      Left = @wfrontc@
#elseif [string compare @dimension@ "1d"] == 0
      Left = 0
#endif
      Right = @wtot@
    )
  )
)
```

Here, the incident light intensity is specified using the keyword `WavePower` and is set to 0 [W/m^2].

The simulation of EQE spectra involves wavelength ramping and solving the Poisson equation, and the electron and hole continuity equations. Before this step, as a convergence aid, these equations are first solved in a `Transient` command, followed by ramping the intensity from `0` to `intensity`:

```
Coupled (Digits = 12){Poisson}
Transient (
  InitialTime = 0
  FinalTime = 1
  InitialStep = 1e-3
  MinStep = 1e-12
  MaxStep = 0.2
  Increment = 2.0
  Decrement = 2.0)
{ Coupled (Iterations = 50) {Poisson Electron Hole} }
Quasistationary (
  InitialStep = 1e-12
  MaxStep = 0.2
  Minstep = 1e-12
  Increment = 1.5
  Goal { Model = "Optics" Parameter = "WavePower"
    value = @intensity@ }
) { Coupled (Iterations = 200)
    {Poisson Electron Hole} }
```

When the intensity increases from `0` to `intensity`, which is `1` in this case, the wavelength ramping is performed in two `Quasistationary` statements in the `Solve` section:

```
Quasistationary (
  DoZero
  InitialStep = 1e-12
  MaxStep = @<1./wsteps>@
  Minstep = 1e-12
  Increment = 2
  Goal {Model = "Optics" Parameter = "Wavelength"
    value = 0.8}
) { Coupled {Poisson Electron Hole} }
Quasistationary (
  DoZero
  InitialStep = 1e-12
  MaxStep = @<1./wsteps/10.>@
  Minstep = 1e-12
  Increment = 2
  Goal {Model = "Optics" Parameter = "Wavelength"
    value = @wend@}
) { Coupled {Poisson Electron Hole} }
```

Here, a finer wavelength ramp is performed after 0.8 µm to resolve the oscillations in the EQE spectra.

The quantum efficiency is calculated in the Inspect node from the integral of the optical generation rate. The

CurrentPlot section is needed to plot output quantities such as integral of the optical generation rate and wavelength of light:

```
CurrentPlot {
  Model = "Optics"
  Parameter = "Wavelength"
  OpticalGeneration(Integrate(Semiconductor))
}
```

Sentaurus Device: Simulating Reflectance Spectrum

The reflectance spectrum of the triple-junction solar cell is simulated in the reflect node of Sentaurus Device of the project 07-MJSC-Subcell-EQE-Reflect.

This simulation involves wavelength ramping and the activation of TMM, as discussed in Sentaurus Device: Simulating EQE Spectra on page 30. Here, the wavelength ramping is performed directly without using the initial transient simulation and intensity ramping.

The keyword WavePower is specified using the Sentaurus Workbench parameter intensity in the Optics section:

```
Optics (
  TMM (
    NodesPerWavelength = 100
    Excitation (
      Theta = 0
      Polarization = 0.5
      Wavelength = @wstart@
      WavePower = @intensity@
    )
  )
)
```

The CurrentPlot section is needed to plot output quantities such as the polarization-dependent reflectance and transmittance as a function of wavelength:

```
CurrentPlot {
  Model = "Optics"
  Parameter = "Wavelength"
}
```

The reflectance is written in the standard Plot file when the wavelength ramp is performed in two Quasistationary statements in the Solve section:

```
Solve {

Optics
Quasistationary (
  InitialStep = @<1./wsteps>@
  MaxStep = @<1./wsteps>@
  Minstep = @<1./wsteps>@
  Goal {Model = "Optics" Parameter = "Wavelength"
    value = 0.8}
) { Optics }
Quasistationary (
  InitialStep = @<1./wsteps/10.>@
  MaxStep = @<1./wsteps/10.>@
  Minstep = @<1./wsteps/10.>@
  Goal {Model = "Optics" Parameter = "Wavelength"
    value = @wend@}
) { Optics }
}
```

Here, a finer wavelength ramp is performed after 0.8 µm to resolve the oscillations in the reflectance spectrum.

Tecplot SV

The Tecplot SV node plots the energy band diagram of the dummy triple-junction solar cell at J_{sc} by using the TDR file nX_Banddgm_Jsc_des.tdr in the Sentaurus Workbench projects 04-MJSC-Subcell-Balance-Jsc and 05-MJSC-Subcell-JV. The energy band diagram at J_{sc} for the triple-junction solar cell is plotted in 06-MJSC-JV and is shown in Figure 21. For details, see Tecplot SV on page 15.

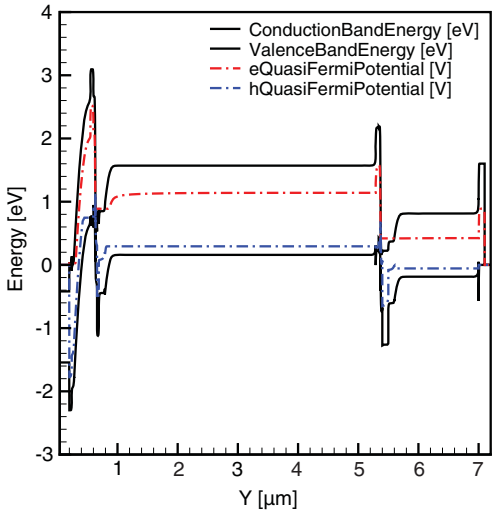

Figure 21 Energy band diagram of triple-junction solar cell at J_{sc}

Inspect: Jsc of Stacked Subcell

Inspect calculates J_{sc} of each stacked subcell in the project 04-MJSC-Subcell-Balance-Jsc. The subcells are current matched to within 10% at J_{sc}.

The subcell J_{sc} values are calculated using the current density values at the electrodes: anode, cathode, toptjContact, and bottjContact. The J_{sc} of the top GaInP subcell, Jsctop, is equal to the cathode current density value and that of the bottom InGaAs subcell, Jscbot, is equal and opposite to the anode current density value. The J_{sc} of the middle GaAs subcell, Jscmid, is computed using Kirchhoff's current law and is equal to the sum of the cathode and toptjContact current densities.

Inspect: Light J–V Characteristics of Stacked Subcells and Triple-Junction Solar Cell

The light J–V curves, P–V curves, and photovoltaic parameters (defined in Eq. 1 to Eq. 3) of all the stacked subcells and triple-junction solar cell are calculated in the projects `05-MJSC-Subcell-JV` and `06-MJSC-JV`, respectively. The data for obtaining the J–V curves is contained in the standard plot files. Figure 9 on page 7 shows the light J–V curves of all the stacked subcells, and Figure 10 on page 7 shows the light J–V and P–V curves of the triple-junction solar cell.

The subcell J–V curves were computed using the method similar to that described in Inspect: Jsc of Stacked Subcell on page 31. The J–V curves of the GaInP and InGaAs subcells were obtained from the cathode and anode voltages and currents, respectively.

Using Kirchhoff's current law, the J–V curve of the middle GaAs subcell was computed by adding the cathode and toptjContact currents as a function of the bottjContact bias voltage:

```
cv_createDS Jbottj($N) "PLT_JV($N) bottjContact
  OuterVoltage" "PLT_JV($N) toptjContact
  TotalCurrent"
cv_createDS Jcathode($N) "PLT_JV($N) bottjContact
  OuterVoltage" "PLT_JV($N) cathode TotalCurrent"
cv_createWithFormula J($N)
  "<Jbottj($N)>+<Jcathode($N)>" A A A A
```

Inspect: Reflectance and EQE Spectra

The reflectance and transmittance coefficients of the triple-junction solar cell and EQE spectra of the stacked subcells are stored in standard plot files and are plotted by the Inspect instance in the `07-MJSC-Subcell-EQE-Reflect` project.

The reflectance spectrum is shown in Figure 11 on page 7. The reflectance and transmittance coefficients are calculated using the thickness of the material layers and the complex refractive index as specified in the `TableODB` section of the parameter file.

Figure 11 also shows the EQE spectra of the stacked subcells. As discussed in Inspect: QE Spectra on page 24, Inspect uses Eq. 7 to calculate EQE spectra. Eq. 7 requires the J_{sc} values of various subcells as a function of wavelength for the calculation of EQE spectra of each subcell. The J_{sc} value for each subcell for various wavelengths is calculated using the method described in Inspect: Jsc of Stacked Subcell on page 31.

References

[1] *Epi/MatPar User Guide*, Mountain View, California: Synopsys, Inc., April 2008.

[2] J. F. Geisz *et al.*, "High-efficiency GaInP/GaAs/InGaAs triple-junction solar cells grown inverted with a metamorphic bottom junction," *Applied Physics Letters*, vol. 91, no. 2, p. 023502, 2007.

[3] M. Hermle *et al.*, "Numerical Simulation of Tunnel Diodes for Multi-junction Solar Cells," *Progress in Photovoltaics: Research and Applications*, vol. 16, no. 5, pp. 409–418, 2008.

[4] ASTM Standard E891-87 (1992) "Tables for Terrestrial Direct Normal Solar Spectral Irradiance Tables for Air Mass 1.5," ASTM International, West Conshohocken, PA (<http://www.astm.org>), February 2009.

[5] A. Luque and S. Hegedus (eds.), *Handbook of Photovoltaic Science and Engineering,* Chichester: John Wiley & Sons, 2003.

[6] W. Guter and A. W. Bett, "*I–V* Characterization of Tunnel Diodes and Multijunction Solar Cells," *IEEE Transactions on Electron Devices*, vol. 53, no. 9, pp. 2216–2222, 2006.

[7] S. M. Sze and K. K. Ng, *Physics of Semiconductor Devices*, USA: Wiley-Interscience, 3rd ed., 2007.

[8] *Template for Creating and Simulating Multilayered Heterostructure Devices with TCAD Sentaurus*, TCAD Sentaurus application note, available from SolvNet at <https://solvnet.synopsys.com/retrieve/023480.html>, February 2009.

[9] J. Piprek, *Semiconductor Optoelectronic Devices: Introduction to Physics and Simulation*, Amsterdam: Academic Press, 2003.

[10] M. Levinshtein, S. Rumyantsev, and M. Shur, *Handbook Series on Semiconductor Parameters, Ternary and Quaternary III-V Compounds*, vol. 2, Singapore: World Scientific, 1999.

[11] *Inspect User Guide*, Version A-2008.09, Mountain View, California: Synopsys, Inc., 2008.

[12] J. Nelson, *The Physics of Solar Cells*, London: Imperial College Press, 2003.

[13] C. Honsberg and S. Bowden, *Photovoltaics: Devices, Systems and Applications*, PV CD-ROM (available from <http://pvcdrom.pveducation.org/>), February 2009.

[14] G. Létay, *Modellierung von III-V Solarzellen*, Ph.D. thesis, Universität Konstanz, Germany, 2003.

Table 1 Details of triple-junction GaInP/GaAs/InGaAs solar-cell structure (2D simulation mode)

Region	Material	SourceParFile	Thickness	Doping	MoleFraction	Refinement
# Layers section:						
cap	GaAs		0.2	−1e19		(mbox 0.01 1.1 both)
#Top InGaP subcell						
topfsf	AlInP		0.04	2e18	0.5	(mbox 0.005 1.1 both)
topem	GaInP		0.04	9e17	0.5	(mbox 0.001 1.5 both)
topbase	GaInP		@dtopbase@	-5e16	0.5	(mbox 0.001 1.5 both)
topbsf	AlInP		0.05	-5e18	0.5	(mbox 0.005 1.5 both)
# Top tunnel diode						
topphighTD	AlGaAs		0.03	-1e19	0.3	(mbox 0.001 1.1 both)
topnhighTD	GaAs		0.03	1e19		(mbox 0.001 1.1 both)
# Middle GaAs subcell						
midfsf	GaInP		0.03	7e18	0.5	(mbox 0.005 1.5 both)
midem	GaAs		0.1	9e17		(mbox 0.001 1.5 both)
midbase	GaAs		@dmidbase@	-5e16		(mbox 0.001 1.5 both)
midbsf	GaInP		0.05	-2e18	0.5	(mbox 0.005 1.5 both)
# Bottom tunnel diode						
botphighTD	AlGaAs		0.03	-1e19	0.3	(mbox 0.001 1.1 both)
botnhighTD	GaAs		0.03	1e19		(mbox 0.001 1.1 both)
# Bottom InGaAs subcell						
botfsf	GaInP	GaInP.tcl	0.1	1e19	0.25	(mbox 0.05 1.2 both)
botem	InGaAs	InGaAs.tcl	0.1	9e17	0.7	(mbox 0.005 1.3 both)
botbase	InGaAs	InGaAs.tcl	@dbotbase@	-5e16	0.7	(mbox 0.01 1.5 both)
botbsf	GaInP	GaInP.tcl	0.1	-1e19	0.25	(mbox 0.001 1.5 both)
# Back contact						
bcon	Gold		0.1			(yref 0.05)
# Material and Interface section:						
# Materialwise parameter file definition						
	AlInP	AlInP.tcl		-5e18	0.5	
	GaInP	GaInP.tcl		1e19	0.5	
	AlGaAs	AlGaAs.tcl		-1e19	0.3	
	GaAs	GaAs.tcl		1e18		
	InGaAs	InGaAs.tcl		-5e16	0.7	
	MgF	MgF.par				
	TiOx	TiOx.par				
# Materialwise interface parameter file definition						
	GaAs/ AlGaAs	GaAs_AlGaAs.par				

Table 2 Details of multilayer stack for Sentaurus Workbench epi project

Region	Material	SourceParFile	Thickness	Doping	MoleFraction	Refinement
# Layers section:						
fsf	AlGaAs	AlGaAs.tcl	0.04	-2.00e18	0.8	(yref 0.01)
emitter	GaAs		0.8	-9.00e17		(mbox 0.05 1.1 both)
base	GaAs		3.2	1.00e17		(mbox 0.01 1.1 both)
bsf	AlGaAs	AlGaAs.tcl	0.2	5.00e18	0.2	(mbox 0.01 1.2 both)
buffer	GaAs		0.35	2.00e18		(mbox 0.01 1.2 both)
# Material and Interface section:						
# Materialwise parameter file definition						
	GaAs	GaAs.tcl		1.00e18		
# Materialwise interface parameter file definition						
	GaAs/ AlGaAs	GaAs_AlGaAs.par				

Table 3 Details of AlInP/GaInP/AlGaAs/GaAs/InGaAs heterostructure

Region	Material	SourceParFile	Thickness	Doping	MoleFraction	Refinement
# Layers section:						
layer1	AlInP	AlInP.tcl	1	1e15	0.5	(mbox 0.001 1.1 both)
layer2	GaInP	GaInP.tcl	1	1e15	0.5	(mbox 0.001 1.1 both)
layer3	AlGaAs	AlGaAs.tcl	1	1e15	0.3	(mbox 0.001 1.1 both)
layer4	GaAs	GaAs.tcl	1	1e15		(mbox 0.001 1.1 both)
layer5	InGaAs	InGaAs.tcl	1	1e15	0.7	(mbox 0.001 1.1 both)

Download Instructions

The Sentaurus Workbench template project `MJ-SolarCell.gzp` can be downloaded.

To download the project:

1. Start an FTP session to `ftp.synopsys.com`, for example:

   ```
   >ftp ftp.synopsys.com
   ```

2. Enter your Synopsys SolvNet® user name.

3. Enter your Synopsys SolvNet password.

4. Type `binary` at the FTP prompt to set the transfer mode to binary:

   ```
   ftp> binary
   ```

5. Type the following commands to obtain the project:

   ```
   ftp> cd labs

   ftp> cd TCAD_Sentaurus_Applications_A-2008.09
   ftp> get MJ-SolarCell.gzp
   ```

6. To log off, type `quit`:

   ```
   ftp> quit
   ```

7. Copy the project to any directory under your Sentaurus database root directory `$STDB`.

8. Double-click the project file in the Sentaurus Workbench projects browser to start the unpacking process.

Assistance with Downloads

If you experience download problems:

- In the USA:
  - Email est-adm@synopsys.com
  - Call the EST Hotline at (650) 584 1631
- In Europe:
  - Call the Synopsys EST Support Center in Ireland: +353 1 436 8880