

Create system-verilog bitmap

במסמך זה תיעוד והנחיות שימוש בפונקציה VerilogBitmapArray.m.
נכתב ע"י רון טייכנר, 25.03.2019

[מטרת ויכולות הפונקציה](#)

[הנחיות שימוש](#)

[כניסות לפונקציה](#)

[דוגמא לאופן הקריאה לפונקציה](#)

[פירוט הכניסות לפונקציה](#)

[מוצא הפונקציה - הדפסות למסך](#)

[הודעות בנושא תכונת שקיפות בתמונה](#)

[הודעת כתיבת קובץ sv](#)

[הודעות שגיאה / אי ביצוע פעולה אפשריות](#)

[מוצא הפונקציה - גרפים](#)

[מוצא הפונקציה - קובץ sv](#)

[ליבת המוצא עבור תמונה בגודל 5x5 קוונטיזציה ל - 8 ביט](#)

[ליבת המוצא עבור תמונה בינארית בגודל 5x5](#)

[שילוב קובץ SV](#)

[דוגמת שימוש](#)

[תכולת הספרייה](#)

מטרת ויכולות הפונקציה

מטרת הפונקציה היא המרה של תמונה מפורמט סטנדרטי של קובץ תמונה אל קובץ בשפת system-verilog.
לאחר המרת התמונה ייכתב לספריית העבודה קובץ system-verilog אותו יש לשלב בפרויקט.

יכולות הפונקציה:

1. תמיכה בפורמטים הבאים: jpg, tif, png.
2. פורמטים נוספים של תמונה נתמכים גם כן, אך רק jpg, tif, png (נבדקו).
3. ביצוע חיתוך (crop) לתמונה לאזור עניין המוגדר ע"י המשתמש.
4. ביצוע שינוי גודל לתמונה לגודל (מספר שורות, מספר עמודות) המוגדר ע"י המשתמש.
5. קביעת רמת שקיפות בינארית עבור תמונות המכילות רמות שקיפות.

הפונקציה מדפיסה למסך את התמונה לאורך שלבי העיבוד השונים וכך למשתמש יש את האפשרות לשחק עם הפרמטרים השונים עד לקבלת התמונה הסופית (שתרשם לקובץ sv) לפי שביעות רצונו.

הנחיות שימוש

כניסות לפונקציה

דוגמא לאופן הקריאה לפונקציה

```
inputImageFileName      = "corn.tif";
outputVerilogFileName    = "cornBitmap.sv";
sProcessing.binaryTransparencyTh = 1; % [%]

sProcessing.sCrop.enable = true;
sProcessing.sCrop.xyPortions = [50,50]; % [%]
sProcessing.sCrop.xyCenter = [50,50]; % [%] % x values are left to
                                     % right; y values are up to down

sProcessing.sResize.enable = true;
sProcessing.sResize.new_xy  = [32,32];

sProcessing.quantize_nBits = 8; % {8 - 3Red, 3Green, 2Blue}
VerilogBitmapArray(inputImageFileName,outputVerilogFileName,sProcessing)
```

פירוט הכניסות לפונקציה

inputImageFileName	מחרוזת - שם קובץ התמונה. על הקובץ להימצא בספריית העבודה או לחלופין ניתן לספק path מלא. דוגמא: "corn.tif" "c:\corn.tif"
outputVerilogFileName	מחרוזת - שם הקובץ אליו יירשם מקטע הקוד בשפת system verilog ובו ה - bitmap של התמונה. דוגמא: "cornBitmap.sv"
sProcessing.sCrop.enable	שדה בולאני, בהינתן שהוא true יבוצע crop לתמונה בהתאם לנתונים בשדות הבאים.
sProcessing.sCrop.xyCenter	וקטור בעל מימדים [1x2].

	<p>הכניסה הראשונה היא המיקום בציר המאוזן המהווה את המרכז סביבו מתבצע החיתוך. היחידות הן אחוזים, כלומר מספר בתחום [0,100].</p> <p>הכניסה השנייה היא המיקום בציר המאונך המהווה את המרכז סביבו מתבצע החיתוך. היחידות הן אחוזים, כלומר מספר בתחום [0,100].</p>
sProcessing.sCrop.xyPortions	<p>וקטור בעל מימדים [1x2].</p> <p>הכניסה הראשונה היא חלק התמונה בציר המאוזן הנשאר בתמונה בעת ביצוע החיתוך. היחידות הן אחוזים, כלומר מספר בתחום [0,100].</p> <p>הכניסה השנייה היא חלק התמונה בציר המאונך הנשאר בתמונה בעת ביצוע החיתוך. היחידות הן אחוזים, כלומר מספר בתחום [0,100].</p>
sProcessing.sResize.enable	<p>שדה בולאני, בהינתן שהוא true יבוצע שינוי גודל (מספר שורות, מספר עמודות) לתמונה (או לתמונה לאחר crop) בהתאם לנתונים בשדות הבאים.</p>
sProcessing.sResize.new_xy	<p>וקטור בעל מימדים [1x2].</p> <p>הכניסה הראשונה היא מספר העמודות בתמונה שתירשם בקובץ sv.</p> <p>הכניסה השנייה היא מספר השורות בתמונה שתירשם בקובץ sv.</p>
sProcessing.quantize_nBits	<p>בשדה זה קובעים את אופן ביצוע בקוונטיזציה לתמונה. אפשריים הערכים הבאים:</p> <p>8 - תיווצר תמונה בעלת הייצוג הבא: [3bitsRed, 3bitsGreen, 2bitsBlue]</p> <p>8 - תיווצר תמונה בעלת הייצוג הבא: [2bitsRed, 1bitsGreen, 1bitsBlue]</p> <p>1 - תיווצר תמונת שחור לבן בינארית</p>
sProcessing.binaryTransparencyTh	<p>סקלר בתחום [0,100].</p> <p>חלק מתמונות בפורמט png תומכות בשקיפות. השקיפות בתמונה הינה ברמה משתנה. ב - system verilog אנו תומכים עבור כל פיקסל בשקיפות בינארית - שקוף לחלוטין או פיקסל התמונה המקורית.</p>

	בשדה זה אנו מכוונים את הסף - רק ערכי שקיפות הגדולים מהסף יהיו שקופים בקובץ sv המיוצר. היחידות הן אחוזים, כלומר מספר בתחום [0,100].
--	--

מוצא הפונקציה - הדפסות למסך

הודעות בנושא תכונת שקיפות בתמונה

הפונקציה מדפיסה למסך הודעה המציינת האם התמונה שהתקבלה מכילה שדה שקיפות. במידה וקיים שדה שקיפות תודפס ההודעה:

```
image has transparency; scale binary threshold using  
sProcessing.binaryTransparencyTh
```

במידה ולא קיים שדה שקיפות תודפס ההודעה:

```
image has no transparency
```

הודעת כתיבת קובץ sv

בסיום פעולה תקינה של הפונקציה תודפס הודעה על כתיבת קובץ לדיסק. לדוגמא:

```
ngc6543aBitmap.sv written to disk
```

הודעות שגיאה / אי ביצוע פעולה אפשריות

במידה ושדה enable של חיתוך / שינוי גודל הוא במצב false תתקבל הודעה:

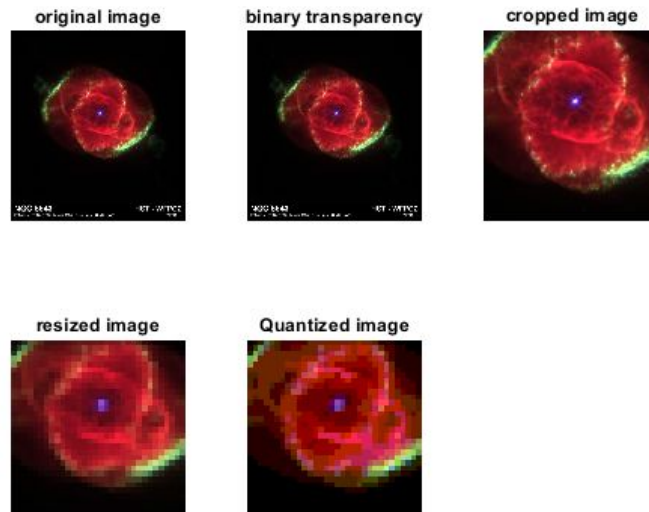
```
sProcessing.sCrop.enable is false - image was not cropped
```

במידה ואחד השדות בקלט חסר תתקבל הודעה בסגנון:

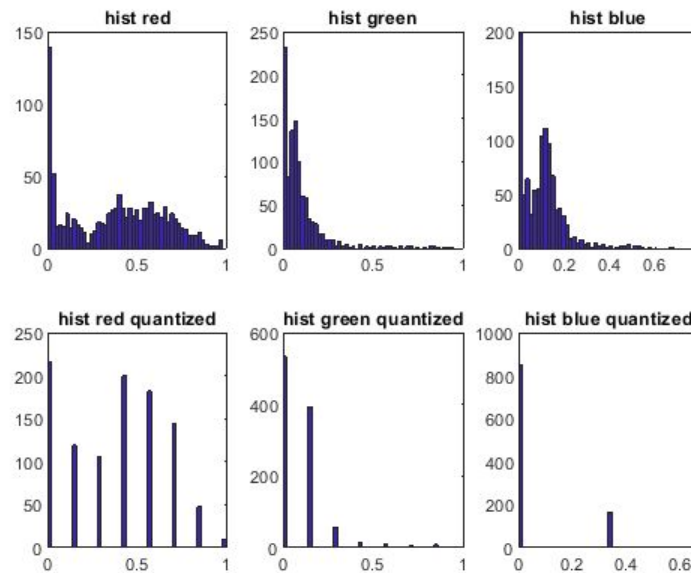
```
sProcessing.sCrop.xyPortions or sProcessing.sCrop.xyCenter are not a  
[2x1] vector - image was not cropped
```

מוצא הפונקציה - גרפים

הפונקציה יוצרת שני גרפים.
הראשון מכיל את התמונה שהתקבלה לאורך העיבוד:



השני מוקדש לנושא הקוונטיזציה ומדגים את פילוג הערכים לכל צבע לפני ואחרי קוונטיזציה:



מוצא הפונקציה - קובץ sv

ממשקי המודול הנוצר:

```
module smileyBitMap (  
    input logic clk,  
    input logic resetN,  
    input logic [10:0] offsetX, // offset from top left position  
    input logic [10:0] offsetY,  
    input logic InsideRectangle, //input that the pixel is within a bracket  
    output logic drawingRequest, //output that the pixel should be displayed  
    output logic [23:0] RGBout //rgb value form the bitmap  
);
```

- בכניסות offsetX, offsetY יש להזין את קואורדינטות הפיקסל המבוקש.
- הכניסה InsideRectangle היא למעשה כניסת אפשר - רק כאשר היא מקבלת ערך 1 לוגי נקבל 1 לוגי ביציאת drawingRequest.
- המוצא drawingRequest הינו העתק של כניסת InsideRectangle כאשר הפיקסל המבוקש אינו שקוף. אחרת מוצא זה הוא 0 לוגי.
- המוצא RGBout הינו שרשרת ביטים ובו 8 ביט לכל צבע.

ליבת המוצא עבור תמונה בגודל 5x5 קוונטיזציה ל - 8 ביט

```
localparam int OBJECT_WIDTH_X = 5;  
localparam int OBJECT_HEIGHT_Y = 5;  
logic [0:OBJECT_HEIGHT_Y-1] [0:OBJECT_WIDTH_X-1] [8-1:0] object_colors = (  
    {8'h44, 8'h64, 8'h84, 8'h84, 8'h20, },  
    {8'h64, 8'h84, 8'h61, 8'h84, 8'h64, },  
    {8'h40, 8'h84, 8'h80, 8'hA4, 8'h85, },  
    {8'h00, 8'h44, 8'h84, 8'hA9, 8'h6D, },  
    {8'h00, 8'h00, 8'h00, 8'h00, 8'h00, }  
);
```

ליבת המוצא עבור תמונה בינארית בגודל 5x5

```
localparam int OBJECT_WIDTH_X = 5;  
localparam int OBJECT_HEIGHT_Y = 5;  
logic [0:OBJECT_HEIGHT_Y-1] [0:OBJECT_WIDTH_X-1] [1-1:0] object_colors = (  
    {1'b0, 1'b0, 1'b0, 1'b0, 1'b0, },  
    {1'b0, 1'b0, 1'b0, 1'b0, 1'b0, },  
    {1'b0, 1'b0, 1'b0, 1'b0, 1'b0, },  
    {1'b0, 1'b0, 1'b0, 1'b0, 1'b0, },  
    {1'b0, 1'b0, 1'b0, 1'b0, 1'b0, }  
);
```

שילוב קובץ SV

שילוב הקובץ הינו זהה לכל אחת מהקוונטיזציות האפשריות.
ממשקי הקובץ לא דורשים שום שינוי.
פשוט מאד לוקחים את הקובץ שהתקבל ומשלבים as-is בפרויקט.

ספרייה זו מכילה דוגמאות לפרויקטים:

1. פרויקט בו משולב קובץ בעל קוונטיזציה ל - 8 ביט:

Lab1Demo-8bit.qar

2. פרויקט בו משולב קובץ בעל קוונטיזציה ל - 4 ביט:

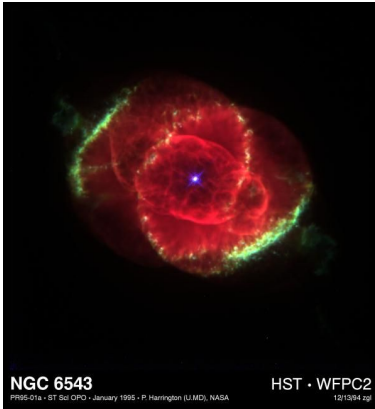
Lab1Demo-4bit.qar

3. פרויקט בו משולב קובץ בעל קוונטיזציה בינארית:

Lab1Demo-1bit.qar

דוגמת שימוש

נרצה לטעון את התמונה, לבצע חיתוך סביב הנקודה המרכזית' להקטין לגודל של [32x32] ולבצע קוונטיזציה ל - 8bit.



הכניסות לפונקציה:

```
inputImageFileName      = "ngc6543a.jpg";
outputVerilogFileName    = "ngc6543aBitmap.sv";

sProcessing.sCrop.enable      = true;
sProcessing.sCrop.xyPortions  = [25,25]; % [%]
sProcessing.sCrop.xyCenter    = [50,45]; % [%]

sProcessing.sResize.enable    = true;
sProcessing.sResize.new_xy    = [32,32];

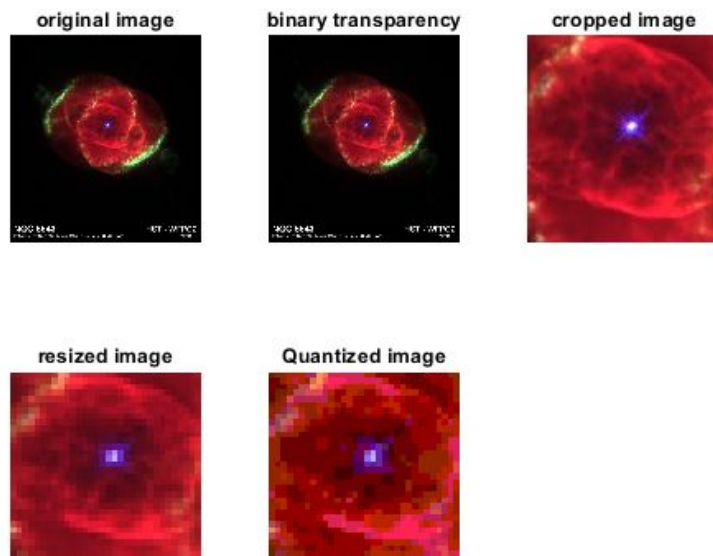
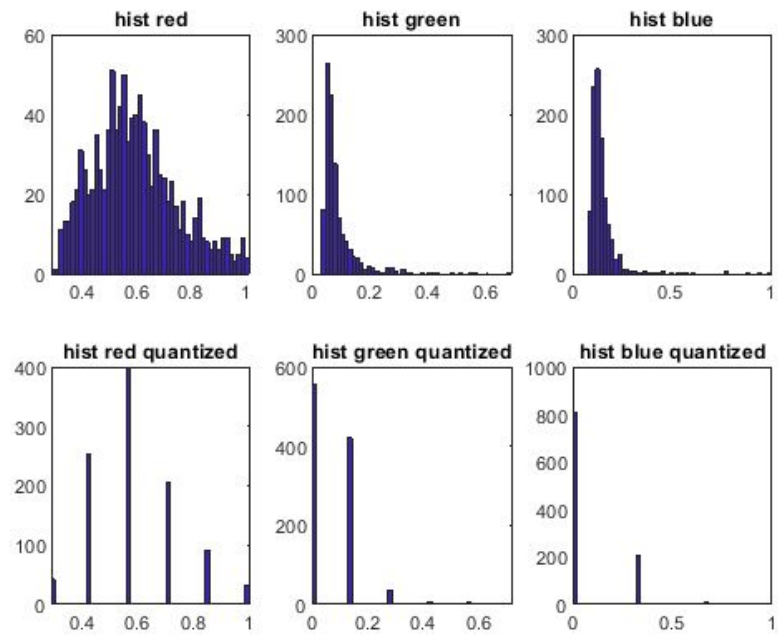
sProcessing.quantize_nBits    = 8;
sProcessing.binaryTransparencyTh = 1; % [%]
```

```
VerilogBitmapArray(inputImageFileName,outputVerilogFileName,sProcessing)
```

פלט הפונקציה למסך:




















```
image has no transparency
ngc6543aBitmap.sv written to disk
```


הגרפים המתקבלים:



תכולת הספרייה

1. הפונקציה VerilogBitmapArray.m
2. דוגמת הרצה בסקריפט RunVerilogBitmapArray_example.m
3. תמונות בפורמטים jpg, png, tif
4. קבצי qar - דוגמאות לפרויקטים
5. מסמך ההדרכה Create system-verilog bitmap.pdf

Current Folder		
	Name ▲	Git
	corn.tif	
	Create system-verilog bitmap.pdf	
	EEEL LOGO TRANSPERANT.png	
	Lab1Demo-1bit.qar	
	Lab1Demo-4bit.qar	
	Lab1Demo-8bit.qar	
	ngc6543a.jpg	
	peppers.png	
	RunVerilogBitmapArray_example.m	
	VerilogBitmapArray.m	