Matrix multiplication - verification

Digital Design and Logic Synthesis

Block: matmul_tb

Digital High Level Design
Version 0.1

By:

Yair Ross 207287889

Dan Bar 212404982

Table of Content

List of figures

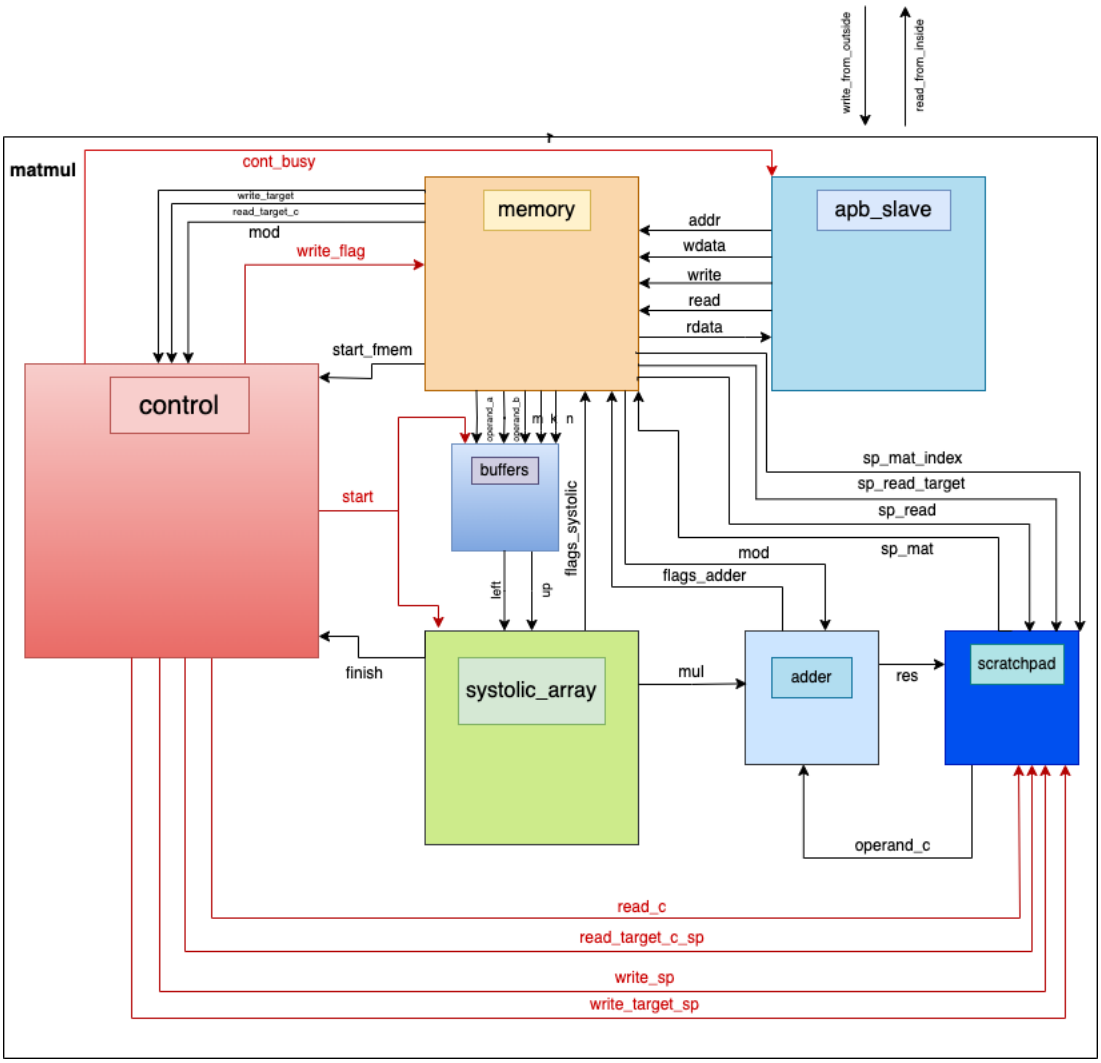


Figure 1: matmul Block Diagram

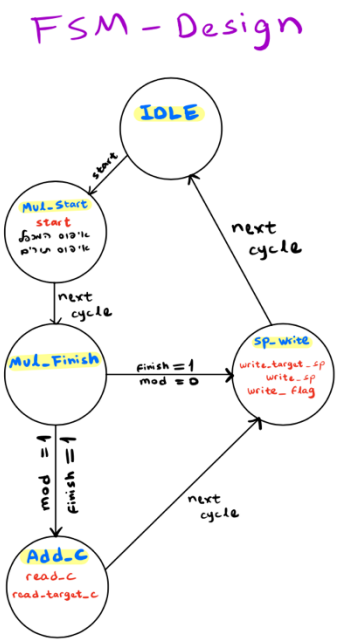


Figure 2: FSM of matmul.

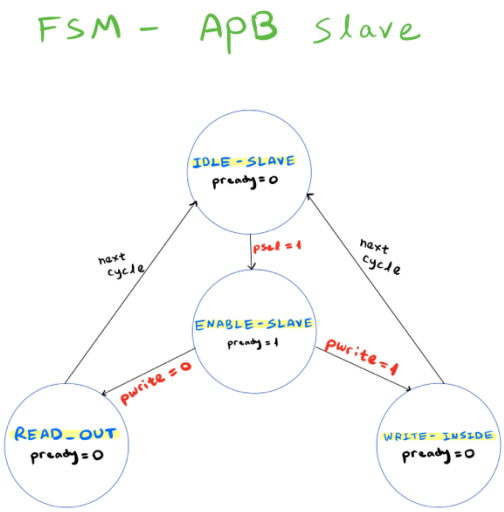


Figure 3: FSM of apb slave

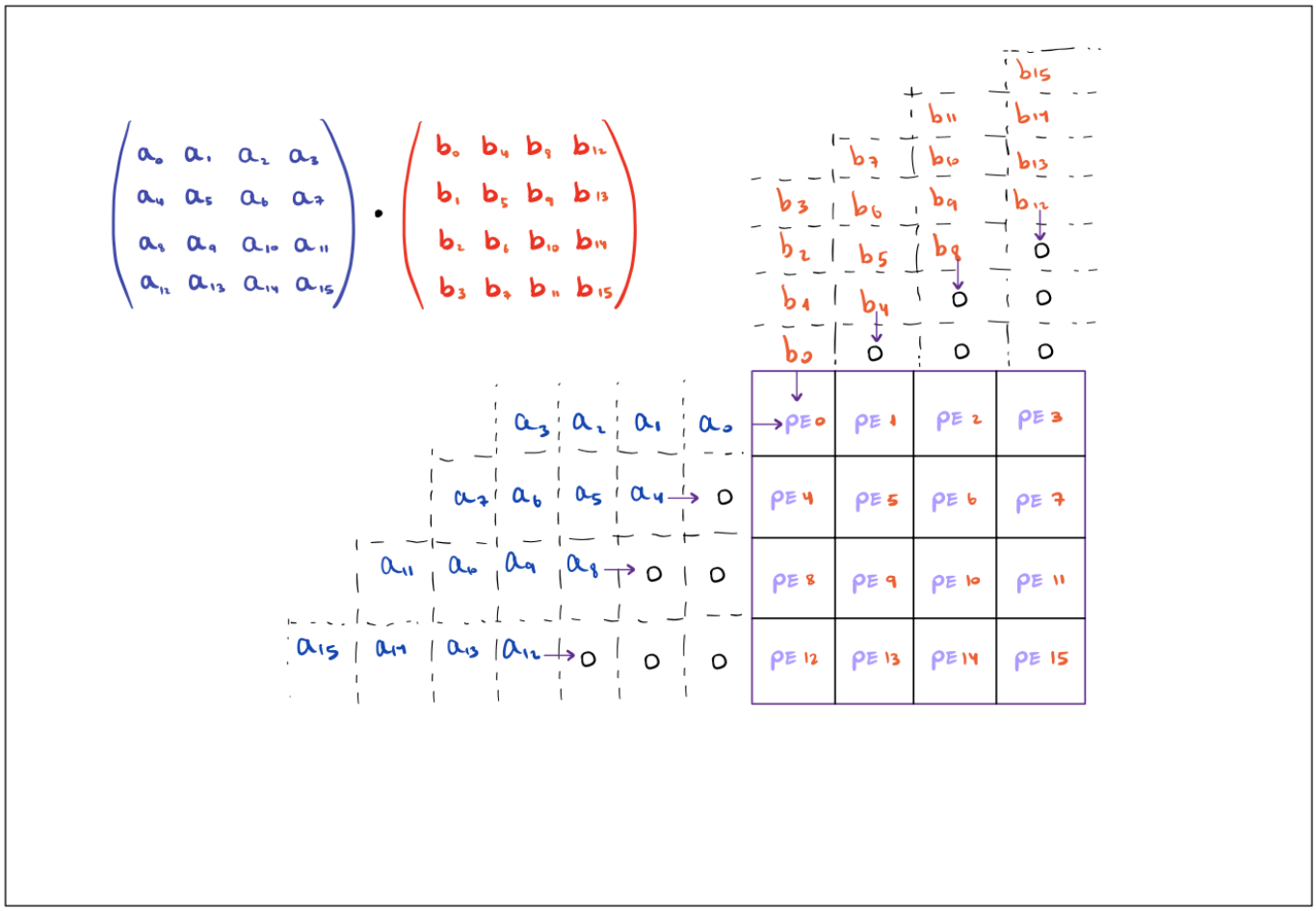


Figure 4: illustration for systolic array / buffers blocks

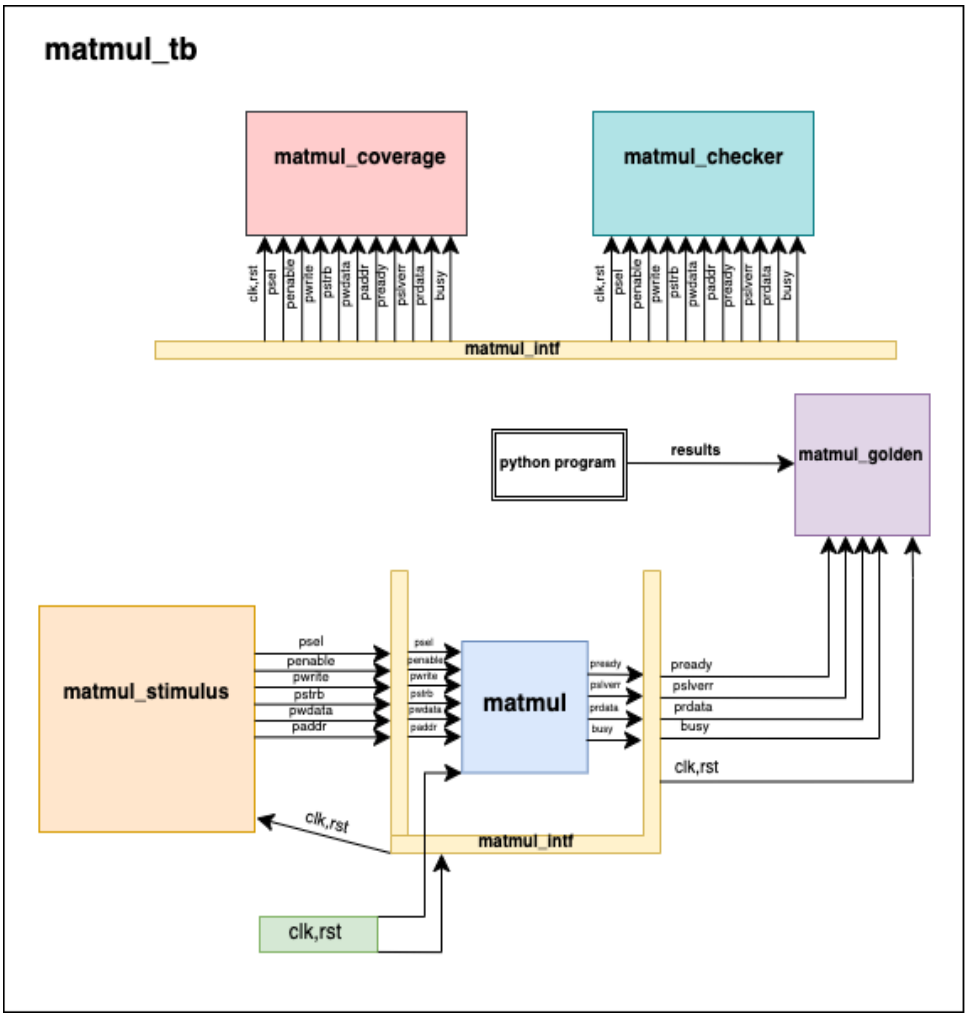


Figure 5: matmul_tb Block Diagram

List of signals tables

פרמטרים בהם תומך מודול ה – Interface:

```
parameter DATA_WIDTH = 16
parameter BUS_WIDTH = 64
parameter ADDR_WIDTH = 16
localparam MAX_DIM = BUS_WIDTH/DATA_WIDTH
parameter SP_NTARGETS = 4
```

כניסות: (ממודול ה – matmul_tb)

```
clk
rst
```

סיגנלים פנימיים:

```
psel // sel
penable // enable
pwrite // write enable
[MAX_DIM-1:0]   pstrb // write strobe (byte select)
[BUS_WIDTH-1:0] pwdata // write data
[ADDR_WIDTH-1:0] paddr // address
pready // slave ready
pslverr // slave error
[BUS_WIDTH-1:0] prdata // read data
busy // Busy signal, indicating the design cannot be written to
```

Figure 1: **matmul_intf** signals table

פרמטרים בהם תומך מודול ה – Stimulus:

```
parameter DATA_WIDTH = 16
parameter BUS_WIDTH = 64
parameter ADDR_WIDTH = 16
localparam MAX_DIM = BUS_WIDTH/DATA_WIDTH
parameter SP_NTARGETS = 4
```

כניסות: (ממודול ה – interface)

```
clk    = s_intf.clk
rst    = s_intf.rst
```

יציאות: (אל מודול ה - interface)

```
clk    = s_intf.clk
rst    = s_intf.rst
psel_o = s_intf.psel
penable_o = s_intf.penable
pwrite_o = s_intf.pwrite
pstrb_o = s_intf.pstrb
paddr_o = s_intf.paddr
pwdata_o = s_intf.pwdata
```

Figure 2: **matmul_stimulus** ports table

פרמטרים בהם תומך מודול ה - Golden – Model :

```
parameter string GOLDEN_PATH = "" // need to be overwritten in test bench
parameter string DUT_PATH = "" // need to be overwritten in test bench
parameter DATA_WIDTH = 16
parameter BUS_WIDTH = 32
parameter ADDR_WIDTH = 16
parameter MAX_DIM = (BUS_WIDTH / DATA_WIDTH)
parameter SP_NTARGETS = 4
```

כניסות: (ממודול ה - interface)

```
clk    = gold_intf.clk
rst    = gold_intf.rst
pready = gold_intf.pready
penable = gold_intf.penable
pwrite = gold_intf.pwrite
psel = gold_intf.psel
prdata = gold_intf.prdata
```

Figure 3: ***matmul_golden*** *ports table*

פרמטרים בהם תומך מודול ה – Coverage:

```
parameter DATA_WIDTH = 16
parameter BUS_WIDTH = 64
parameter ADDR_WIDTH = 16
localparam MAX_DIM = BUS_WIDTH/DATA_WIDTH
parameter SP_NTARGETS = 4
```

כניסות: (ממודול ה - interface)

```
clk    = coverage_bus.clk
rst    = coverage_bus.rst
pready = coverage_bus.pready
penable = coverage_bus.penable
pwrite = coverage_bus.pwrite
psel = coverage_bus.psel
prdata = coverage_bus.prdata
pstrb = coverage_bus.pstrb
paddr = coverage_bus.paddr
pwdata = coverage_bus.pwdata
busy = coverage_bus.busy
pslverr = coverage_bus.pslverr
```

Figure 4: ***matmul_coverage*** *ports table*

פרמטרים בהם תומך מודול ה – Checker:

```
parameter DATA_WIDTH = 16
parameter BUS_WIDTH = 64
parameter ADDR_WIDTH = 16
localparam MAX_DIM = BUS_WIDTH/DATA_WIDTH
parameter SP_NTARGETS = 4
```

כניסות: (ממודול ה - interface)

```
clk     = checker_bus.clk
rst     = checker_bus.rst
pready =  checker_bus.pready
penable = checker_bus.penable
pwrite = checker_bus.pwrite
psel = checker_bus.psel
prdata = checker_bus.prdata
pstrb = checker_bus.pstrb
paddr = checker_bus.paddr
pwdata = checker_bus.pwdata
busy = checker_bus.busy
pslverr = checker_bus.pslverr
```

Figure 5: ***matmul_checker*** *ports table*

פרמטרים בהם תומך מודול ה – tb:

כניסות: (אל מודול ה – interface) ו - (אל מודול ה – matmul)

```
clk
rst
```

Verification Plan

מטרת הוריפיקציה היא לבחון את המערכת שעיצבנו בחלק הראשון של הפרויקט ולתקן אותה במידת הצורך.

כעת רכיב ה – matmul שכתבנו יהפוך ל- dut (device under test) ונכניס אותו כאחד המודולים בתוך מודול מרכזי הנקרא matmul_tb.

נגדיר את מודול ה - matmul_tb כגנרטור של clock,reset למודול ה – interface ומודול ה – matmul.

מודול ה – interface הוא חבילה שעוטפת את כל הפורטים של מודול ה – matmul ומאפשרת גישה למודולים האחרים שכתבנו לגשת אל פורטים אלו בצורה נוחה.

במסגרת הבדיקות מטרתנו הינה לשמש למודול ה – matmul , כמאסטר שמזין נתונים אל ה – slave ומקבל תוצאות חישוב ממנו.

הבדיקות שנעשה למערכת הן בדיקות כפל מטריצות וחיבור מטריצות על פי דרישות המשימה והשוואת התוצאות אל תוצאות של קוד פייתון שביצע את אותו החישוב ב – high level.

במסגרת הבדיקות ניצור קובץ של נתונים בצורה רנדומית (בעזרת קוד פייתון הנקרא generate_random_big.py)

ונזין אותו כקלט אל ה – stimulus וקוד הפייתון (golden_model_big.py).

פלט המערכת שהגיע מה – dut נשמר גם הוא בקובץ כמו גם תוצאות החישוב של קוד הפייתון.

ע"י מודול ה – matmul_golden נשווה בין קבצי התוצאות והרמנו דגל hit כאשר הפלט זהה ו – miss אחרת.

ננסה להזין כמה שיותר בדיקות רנדומליות עבור מקרים שונים ובנוסף נזין בדיקות נוספות כגון: בדיקות לכתיבת עם strobe וקריאת דגלים.

Verification Test Objectives

המטרה של ה – test bench הינה לוודא שכל הפלטים של המערכת הינם נכונים.

עושים זאת ע"י הזנה של קובץ inputs למערכת שבו בשורה הראשונה יש את הפרמטרים של ה – control register

ובארבע שורות הבאות (המטריצות שעליהן ה – test bench נבדק הן מטריצות בגודל 4*4 עם פרמטרים

(BUS_WIDTH = 32, DATA_WIDTH = 16 נמצאים איברי המטריצות.

ה – test bench מבצע פעולות כפל וחיבור לפי הקלט ולאחר ביצוע פעולות אלה התוצאות נקראות מתוך ה – dut

ומשוות עם התוצאות של ה – golden model.

לאחר שפעולות כפל וחיבור רבות מבוצעות, אנו בודקים בנוסף כפל וחיבור של מטריצות לא ריבועיות וזאת במטרה למתוח את

המערכת. כמו כן אנו מבצעים בדיקה לתקינות ה – strobe ובדיקה לנכונות הדגלים – flags.

במהלך פעולות הכפל והחיבור אנו ננסה לכתוב למטריצות A,B בכדי לבדוק את נכונות ה – pslverr.

Test Bench Architecture

ה – test bench מורכב ממודול מרכזי בשם matmul_tb שתפקידו להזין את כלל המערכת בשעון ואתחול.

ישנם מספר מודולים בתוך ה – test bench שכל אחד מהם תופס תפקיד הכרחי בבדיקות.

מודול ה – interface -

חבילה שעוטפת את כל הפורטים של מודול ה – matmul ומאפשרת גישה למודולים האחרים שכתבנו

לגשת אל פורטים אלו בצורה נוחה.

מודול ה – stimulus –

או בשמו matmul_stimulus משמש כגנרטור למערכת ומזין את ה – dut דרך ה – slave בערכים כמו

אופרנדי חישוב A,B , וקטור control וסיגנלי בקרה כמו psel,penable,pwrite המגדירים את סוג האופרציה הנבחרת

(קריאה/כתיבה) ואת תזמון האופרציה.

בתחילה המודול מבצע אתחול למערכת (reset).

לאחר מכן קורא מקובץ נתונים את פרמטרי ה – control register עליהן נפעיל את האופרציה ואת המטריצות ושומר אותן

במשתנים לוקאליים.

בהמשך מזין ה – stimulus ל –dut את המטריצות A,B ואת וקטור ה – control הכולל את ה – start bit ואת מימדי המטריצות עליהן נפעיל את האופרציה.

לאחר מספר מחזורי שעון בהם ממתין הגנרטור לפלט המערכת מה – dut מזין לו אותות קריאה לקבלת הפלט בסיגנל prdata.

ה – stimulus מדמה קריאה וכתיבה עם פרוטוקול AMBA.

מודול ה – golden model –

מודול זה נועד להשוואה בין פלט ה – dut לבין פלט קוד הפייתון שמבצע אותם חישובים.

בתחילה המודול קורא מקובץ תוצאה של קוד הפייתון מטריצת תוצאה אל מערך מקומי.

נאפס משתנים לוקאליים hit,miss שנותנים אינדיקציה על חישוב מדוייק או על כשלון.

במהלך הריצה ה – golden model קורא את תוצאות ה – dut , משווה אותן אל תוצאות קוד הפייתון.

עבור משתנים זהים נעלה דגל hit ועבור מחרוזות שונות נעלה דגל miss.

נשמור את תוצאות ה – dut אל קובץ matmul_results.txt.

מודול ה – functional coverage –

מודול זה מקבל מה – interface את כל פורטי הכניסה והיציאה של ה – apb slave

ומטרתו היא לבדוק האם הגענו לכלל הערכים הרלוונטיים עבור כל סיגנל בזמן הסימולציה.

בכתיבת מודול זה רצינו לראות שאין ערכים של ה – slave שלא מתקבלים מסיבות כאלה ואחרות.

חשוב להדגיש כי בדקנו את ערכי הכניסה ל – slave וגם את ערכי היציאה כדי לקבל תמונה גדולה של הסיגנלים.

בעת הרצת הסימולציה נקבל דוח coverage שנותן פירוט על איזה ערכים הגיעו הסיגנלים ולאיזה ערכים לא.

ניתן לראות בהמשך בטבלה ייעודית את טבלת הבדיקות שביצענו במסגרת מודול זה.

מודול ה – functional checker -

מודול זה בודק תרחישים בתקשורת עם ה – apb slave.

תרחישים לדוגמא הם – האם כאשר מתבצע אתחול => סיגנלים הרצויים אכן מתאפסים בזמן יעודי.

בדיקות נוספות שבדקנו הן – האם לאחר שסיגנל pready עולה ל – 1 , הוא גם יורד במחזור שעון הבא כדרוש.

כאשר בדיקה לא מתקיימת נכתוב הערת סימולציה המתחילה ב $ שמעידה על סוג התקלה או סוג הבדיקה שנפלה.

נשים לב כי בעת ההרצה של הסימולציה בתוכנת ה – questasim נקבל דוח מפורט על הבדיקות שנכשלו והצליחו.

עבור כישלון נקבל הודעת שגיאה ועבור הצלחה נקבל עלייה במונה בשם pass.

ניתן לראות בהמשך בטבלה ייעודית את טבלת הבדיקות שביצענו במסגרת מודול זה.

Functional Coverage Table

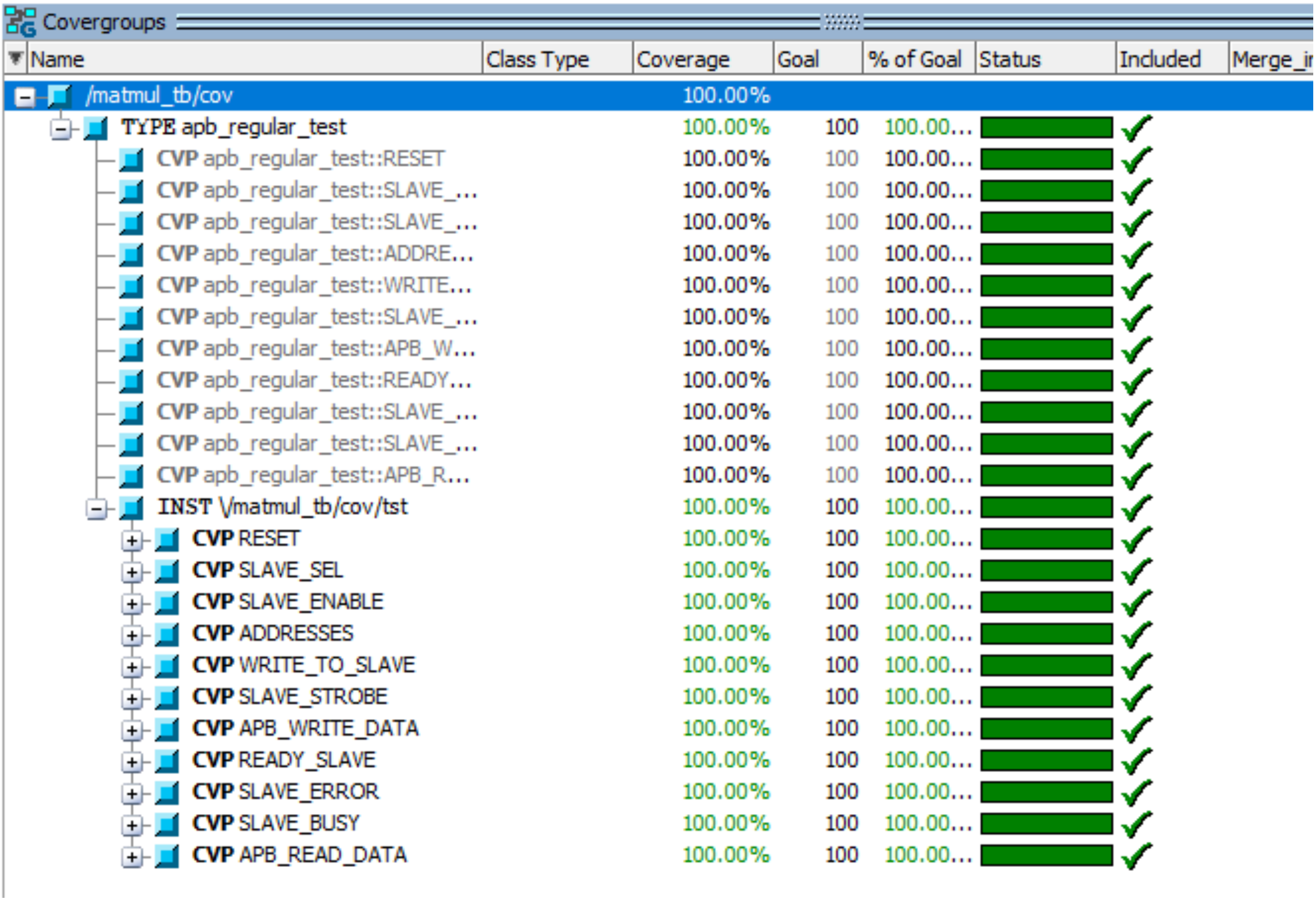
FUNCTION	EVENT	COVERAGE POINT	BINS	scenario
RESET	Posedge clock	rst	0,1	standard
SLAVE_SEL	Posedge clock	psel	0,1	Standard
SLAVE_ENABLE	Posedge clock	penable	0,1	Standard
ADDRESSES	Posedge clock	paddr	0,4,8,12,(16->28)	Standard
WRITE_OR_READ_ TO_SLAVE	Posedge clock	pwrite	0,1	Standard
SLAVE_STROBE	Posedge clock	pstrb	0-> 2**MAX_DIM-1	Standard
APB_WRITE_DATA	Posedge clock	pwdata	0->2**BUS_WIDTH-1	Standard
READY_SLAVE	Posedge clock	pready	0,1	Standard
SLAVE_ERROR	Posedge clock	pslverr	0,1	Standard
SLAVE_BUSY	Posedge clock	busy	0,1	Standard
APB_READ_DATA	Posedge clock	prdata	0->2**BUS_WIDTH-1	Standard

Functional Checker Table

Condition	Expected Result	Scenario
Reset Active (rst=0)	pready → 0 prdata → 0 pslverr → 0 busy → 0	Standard
read_operation_is_execute (psel=1 && penable=1 && pwrite=0)	pready (##1) → 1 || pslverr → 1	Standard
write_operation_is_execute (psel=1 && penable=1 && pwrite=1)	pready (##1) → 1 || pslverr → 1	Standard
pready_transition (pready=1)	pready (##1) → 0	Standard
busy_bit (pready=1 && pwdata[0]=1 && paddr = 0)	busy (##1)→ 1	Standard
Valid output (pready=1 && pwrite=0)	prdata → [0:2**BUS_WIDTH-1)]	Standard
Valid error (pslverr = 1)	pready → 0	Extreme

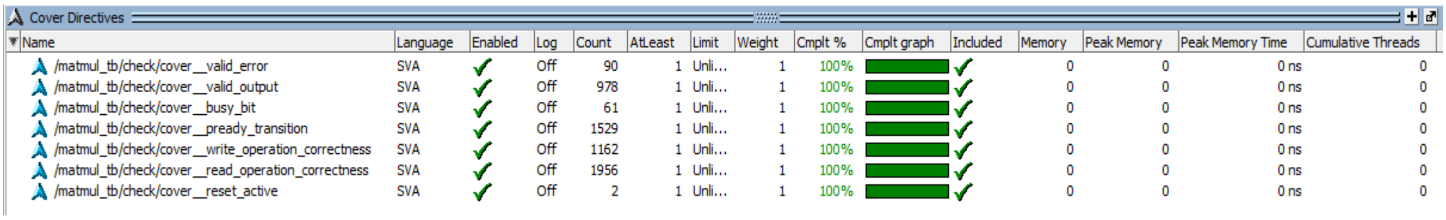
Functional coverage and checker results:

ניתן לראות שעברנו על ה – bins הרצויים:



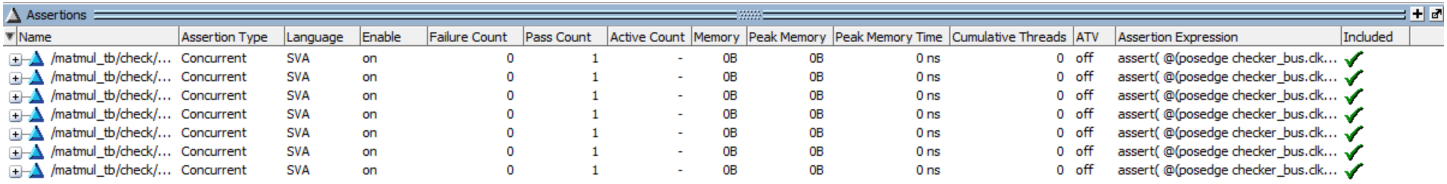
Covergroups

Name	Class Type	Coverage	Goal	% of Goal	Status	Included	Merge_i
/matmul_tb/cov		100.00%					
TYPE apb_regular_test		100.00%	100	100.00…	🟩	✓	
CVP apb_regular_test::RESET		100.00%	100	100.00…	🟩	✓	
CVP apb_regular_test::SLAVE_…		100.00%	100	100.00…	🟩	✓	
CVP apb_regular_test::SLAVE_…		100.00%	100	100.00…	🟩	✓	
CVP apb_regular_test::ADDRE…		100.00%	100	100.00…	🟩	✓	
CVP apb_regular_test::WRITE…		100.00%	100	100.00…	🟩	✓	
CVP apb_regular_test::SLAVE_…		100.00%	100	100.00…	🟩	✓	
CVP apb_regular_test::APB_W…		100.00%	100	100.00…	🟩	✓	
CVP apb_regular_test::READY…		100.00%	100	100.00…	🟩	✓	
CVP apb_regular_test::SLAVE_…		100.00%	100	100.00…	🟩	✓	
CVP apb_regular_test::SLAVE_…		100.00%	100	100.00…	🟩	✓	
CVP apb_regular_test::APB_R…		100.00%	100	100.00…	🟩	✓	
INST \/matmul_tb/cov/tst		100.00%	100	100.00…	🟩	✓	
CVP RESET		100.00%	100	100.00…	🟩	✓	
CVP SLAVE_SEL		100.00%	100	100.00…	🟩	✓	
CVP SLAVE_ENABLE		100.00%	100	100.00…	🟩	✓	
CVP ADDRESSES		100.00%	100	100.00…	🟩	✓	
CVP WRITE_TO_SLAVE		100.00%	100	100.00…	🟩	✓	
CVP SLAVE_STROBE		100.00%	100	100.00…	🟩	✓	
CVP APB_WRITE_DATA		100.00%	100	100.00…	🟩	✓	
CVP READY_SLAVE		100.00%	100	100.00…	🟩	✓	
CVP SLAVE_ERROR		100.00%	100	100.00…	🟩	✓	
CVP SLAVE_BUSY		100.00%	100	100.00…	🟩	✓	
CVP APB_READ_DATA		100.00%	100	100.00…	🟩	✓	



Cover Directives

Name	Language	Enabled	Log	Count	AtLeast	Limit	Weight	Cmplt %	Cmplt graph	Included	Memory	Peak Memory	Peak Memory Time	Cumulative Threads
/matmul_tb/check/cover__valid_error	SVA	✓	Off	90	1	Unli…	1	100%	🟩	✓	0	0	0 ns	0
/matmul_tb/check/cover__valid_output	SVA	✓	Off	978	1	Unli…	1	100%	🟩	✓	0	0	0 ns	0
/matmul_tb/check/cover__busy_bit	SVA	✓	Off	61	1	Unli…	1	100%	🟩	✓	0	0	0 ns	0
/matmul_tb/check/cover__pready_transition	SVA	✓	Off	1529	1	Unli…	1	100%	🟩	✓	0	0	0 ns	0
/matmul_tb/check/cover__write_operation_correctness	SVA	✓	Off	1162	1	Unli…	1	100%	🟩	✓	0	0	0 ns	0
/matmul_tb/check/cover__read_operation_correctness	SVA	✓	Off	1956	1	Unli…	1	100%	🟩	✓	0	0	0 ns	0
/matmul_tb/check/cover__reset_active	SVA	✓	Off	2	1	Unli…	1	100%	🟩	✓	0	0	0 ns	0



Assertions

Name	Assertion Type	Language	Enable	Failure Count	Pass Count	Active Count	Memory	Peak Memory	Peak Memory Time	Cumulative Threads	ATV	Assertion Expression	Included
/matmul_tb/check/…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge checker_bus.clk…	✓
/matmul_tb/check/…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge checker_bus.clk…	✓
/matmul_tb/check/…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge checker_bus.clk…	✓
/matmul_tb/check/…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge checker_bus.clk…	✓
/matmul_tb/check/…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge checker_bus.clk…	✓
/matmul_tb/check/…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge checker_bus.clk…	✓
/matmul_tb/check/…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge checker_bus.clk…	✓

Coverage Report Summary Data by file

```
================================================================================
=== File: C:/Users/rossy/Downloads/212404982_207287889/212404982_207287889/matrix_multiplexer_lib/hdl/adder.v
================================================================================
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                         2         2         0   100.00%
    Expressions                      3         3         0   100.00%
    Statements                       5         5         0   100.00%
    Toggles                         34         2        32     5.88%

================================================================================
=== File: C:/Users/rossy/Downloads/212404982_207287889/212404982_207287889/matrix_multiplexer_lib/hdl/apb_slave.v
================================================================================
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                         15        14         1    93.33%
    Conditions                        3         3         0   100.00%
    Expressions                      12         5         7    41.66%
    FSM States                        4         4         0   100.00%
    FSM Transitions                   6         6         0   100.00%
    Statements                       30        29         1    96.66%
    Toggles                         364       326        38    89.56%

================================================================================
=== File: C:/Users/rossy/Downloads/212404982_207287889/212404982_207287889/matrix_multiplexer_lib/hdl/aritmetic_block.v
================================================================================
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                          4         4         0   100.00%
    Expressions                       4         0         4     0.00%
    Statements                       16        16         0   100.00%
    Toggles                         330       325         5    98.48%

================================================================================
=== File: C:/Users/rossy/Downloads/212404982_207287889/212404982_207287889/matrix_multiplexer_lib/hdl/buffers.v
================================================================================
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                         64        54        10    84.37%
    Conditions                       96        20        76    20.83%
    Statements                      151       141        10    93.37%
    Toggles                         722       654        68    90.58%
```

```
================================================================================
=== File: C:/Users/rossy/Downloads/212404982_207287889/212404982_207287889/matrix_multiplexer_lib/hdl/control.v
================================================================================
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                         14        13         1    92.85%
    FSM States                        5         5         0   100.00%
    FSM Transitions                   9         6         3    66.66%
    Statements                       28        26         2    92.85%
    Toggles                         148        82        66    55.40%

================================================================================
=== File: C:/Users/rossy/Downloads/212404982_207287889/212404982_207287889/matrix_multiplexer_lib/hdl/matmul.v
================================================================================
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                       1184      1078       106    91.04%

================================================================================
=== File: C:/Users/rossy/Downloads/212404982_207287889/212404982_207287889/matrix_multiplexer_lib/hdl/matmul_coverage.sv
================================================================================
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                       1         1         0   100.00%

================================================================================
=== File: C:/Users/rossy/Downloads/212404982_207287889/212404982_207287889/matrix_multiplexer_lib/hdl/matmul_golden.sv
================================================================================
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                         10         7         3    70.00%
    Conditions                        8         3         5    37.50%
    Statements                       22        16         6    72.72%
    Toggles                         576       103       473    17.88%

================================================================================
=== File: C:/Users/rossy/Downloads/212404982_207287889/212404982_207287889/matrix_multiplexer_lib/hdl/matmul_intf.sv
================================================================================
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         184       165        19    89.67%
```

```
================================================================================
=== File: C:/Users/rossy/Downloads/212404982_207287889/212404982_207287889/matrix_multiplexer_lib/hdl/matmul_stimulus.sv
================================================================================
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                        12         5         7    41.66%
    Conditions                       4         0         4     0.00%
    Statements                     155       151         4    97.41%
    Toggles                        722       131       591    18.14%

================================================================================
=== File: C:/Users/rossy/Downloads/212404982_207287889/212404982_207287889/matrix_multiplexer_lib/hdl/matmul_tb.sv
================================================================================
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                       7         7         0   100.00%
    Toggles                          4         3         1    75.00%

================================================================================
=== File: C:/Users/rossy/Downloads/212404982_207287889/212404982_207287889/matrix_multiplexer_lib/hdl/memory.v
================================================================================
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                        36        32         4    88.88%
    Conditions                      15        14         1    93.33%
    Statements                      54        51         3    94.44%
    Toggles                       1064       804       260    75.56%

================================================================================
=== File: C:/Users/rossy/Downloads/212404982_207287889/212404982_207287889/matrix_multiplexer_lib/hdl/scratchpad.v
================================================================================
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                        24        20         4    83.33%
    Conditions                       8         8         0   100.00%
    Statements                      41        31        10    75.60%
    Toggles                        222        95       127    42.79%

================================================================================
=== File: C:/Users/rossy/Downloads/212404982_207287889/212404982_207287889/matrix_multiplexer_lib/hdl/systolic_array.v
================================================================================
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                         6         6         0   100.00%
    Conditions                       1         1         0   100.00%
    Statements                       9         9         0   100.00%
    Toggles                        176       143        33    81.25%


TOTAL COVERGROUP COVERAGE: 100.00%  COVERGROUP TYPES: 1

TOTAL DIRECTIVE COVERAGE: 100.00%  COVERS: 7

TOTAL ASSERTION COVERAGE: 100.00%  ASSERTIONS: 7

Total Coverage By File (code coverage only, filtered view): 71.95%
```

ניתן לראות שרוב הקוד שלנו מכוסה. (לפחות עבור מודולי ה – dut).

אך יש מקרים שפספסנו חלקים בקוד. (לדוגמא הכפלת מטריצות בכל המימדים ואפשרויות אחרות נוספות).

ניתן לתקן זאת ע״י בדיקת יותר מקרים ויותר מקרי קצה אך לא הספקנו לממש יותר בדיקות עבור מקרים נוספים.

לעומת זאת עבור הקודים של ה – stimulus וה – golden הקוד מכוסה פחות אך זה לא בעיה כזו גדולה שכן יש שם חלקים גדולים שממילא אין צורך להיכנס אליהם.

Golden Model (python code)

מטרתינו בכתיבת הקוד הייתה לבחון את הפעולות אותן נדרשנו לממש ב –low level .

נשים לב כי בדרישות נדרשנו לבצע מכפלה בין מטריצות A ,B ולשמור את התוצאה ב – scratchpad.

בנוסף נדרשנו לבצע פעולת חיבור בין שתי תוצאות מכפלות A,B שונים ולשמור את גם תוצאה זו ב – scratchpad.

קוד פייתון זה מקבל קובץ טקסט של נתוני מטריצות.

בקוד נקראים נתוני המטריצות ומתבצעת מכפלה בינהן באמצעות ספריית numpy.

תוצאת המכפלה נשמרת בקובץ מוצא.

לאחר מכן מתבצעת פעולה זו בשנית על מטריצות שונות כאשר גם התוצאה הזו נשמרת באותו הקובץ.

כעת נקראות שתי מטריצות התוצאה שזה עתה חישבנו ושמרנו מאותו קובץ תוצאות ומתבצעת בינהן פעולת חיבור.

גם תוצאת פעולת החיבור נשמרת באותו קובץ תוצאות.

פעולה זו מתבצעת הרבה פעמים על הרבה מטריצות שונות.

תוצאות החישוב נועדו להשוואה מהירה ופשוטה עם תוצאות הדיזיין על אותם חישובים בדיוק.

Golden model (matmul_golden) results:

```
VSIM 1> run
# finished with          978 hits and          0 misses and          0 errors
```