

עמוס זסלבסקי

תֵּכּוֹן סִפְרָתִי וּמְבּוֹא לִפְרוֹיִיקָטִים

עם רכיבי Altera



הוצאת שורש (אלי מיטב) 052 – 2671210

email: elmtv@netvision.net.il

web: <http://shoresh.sfarim.net>

תוכן עניינים

| עמוד | תת-נושא | פרק |
|-----------------|--|----------|
| 1 ----- | פרק המבוא | |
| 2 ----- | מבוא לטכנולוגיה אלקטרונית של רכיבים מתוכנתים | |
| 3 ----- | האבולוציה של כרטיסי תרגול לרכיבים מתוכנתים של Altera | |
| 6 ----- | ייעודו של הספר | |
| 7 ----- | מבנה הספר והמלצות ללומד | |
| 11 ----- | כיצד פותחו החומרים שבספר | |
| 12 ----- | תודות | |
| 15 ----- | הכרת Quartus ותכן גרפי בסיסי | 1 |
| 16 ----- | מבוא | |
| 16 ----- | הפעלת הכלי | |
| 18 ----- | יצירת פרויקט ובחירת הרכיב | |
| 26 ----- | תיאור המערכת שתמומש וצורת החיווט שלה ללוח התרגול | |
| 30 ----- | תיאור המערכת שתמומש באמצעות שרטוט | |
| 35 ----- | שמירת קובץ והוספתו לפרויקט | |
| 36 ----- | קומפילציה של הפרויקט | |
| 40 ----- | עריכת קובץ הקצאת הדקים | |
| 44 ----- | ביצוע הקצאת ההדקים | |
| 48 ----- | סימולציה פנימית בסביבת Quartus | |
| 59 ----- | הבדיקה - באמצעות לוח התרגול | |
| 61 ----- | קונפיגורציה (תכנות/צריבה) של הרכיב | |
| 67 ----- | תרגילי תכן נוספים | |
| 69 ----- | שימוש בלוחות תרגול אחרים | |
| 71 ----- | הכרת מונים ב - Quartus | 2 |
| 72 ----- | מבוא | |
| 73 ----- | הכרת מונים סינכרוניים קלאסיים | |
| 77 ----- | חיווט מונה בינארי 74161 | |
| 79 ----- | ביצוע סימולציה למונה | |
| 82 ----- | בדיקת מונה BCD - 74160 | |
| 82 ----- | יצירת מונה שנתקע באמצעות שימוש ביציאה TC ובכניסות האפשרור | |
| 86 ----- | שימוש ב - 74169 לקבלת מונה בינארי שנתקע בקצותיו ושמוסוגל לשנות כיוון | |
| 88 ----- | הבעייתיות של יצירת מונים גדולים | |
| 89 ----- | יצירת מונה גמיש באמצעות שימוש ב - MegaWizard | |
| 91 ----- | קבלת עזרה על הרכיב הגמיש LPM_COUNTER | |
| 93 ----- | המשך יצירת המונה הגמיש באמצעות שימוש ב - MegaWizard | |
| 99 ----- | יצירת מונה 16 ביט ממונה גמיש באופן גרפי ישיר | |
| 104 ----- | תרגיל תכן | |

| עמוד | תת-נושא | פרק |
|------|---|-----|
| 105 | ----- Quartus – רגיסטרים ב-3 | 3 |
| 106 | ----- מבוא | |
| 109 | ----- רגיסטרים ב- Quartus | |
| 112 | ----- יצירת Ring-Counter | |
| 113 | ----- שימוש ב- LPM_SHIFTREG ואתחול סינכרוני של המונה | |
| 114 | ----- שימוש באתחול א-סינכרוני עם DFF חיצוני | |
| 115 | ----- שימוש ביכולת האתחול הא-סינכרונית של LPM_SHIFTREG | |
| 115 | ----- שימוש באתחול סינכרוני עצמי | |
| 118 | ----- יצירת רגיסטר הזזה ומונה טבעת דו כיווניים | |
| 119 | ----- יצירת מונה טבעת מפותל | |
| 121 | ----- תרגילי תכן נוספים | |
| 123 | ----- תכן סינכרוני | 4 |
| 124 | ----- מבוא לתכן סינכרוני | |
| 124 | ----- תיאור תכן סינכרוני בשתי צורות: ברמת RTL וכמכונת מצבים | |
| 127 | ----- חישוב תדר השעון המכסימלי של תכן סינכרוני | |
| 128 | ----- הצורך בקיום זמן הכנה פנימי במכונה | |
| 129 | ----- חישוב זמן ההכנה של הכניסות וזמן ההחזקה של הכניסות במערכת סינכרונית | |
| 130 | ----- חישוב זמני השהייה של מערכת סינכרונית | |
| 131 | ----- כיצד מקבלים באופן מעשי מידע על התזמונים של המערכת | |
| 133 | ----- טכניקה להגדלת תדר שעון של המערכת שמבוססת על PipeLine | |
| 135 | ----- יציאות "מלוכלכות" וטכניקות בסיסיות ליצירת תכן סינכרוני | |
| 141 | ----- טכניקות לניקוי יציאות | |
| 143 | ----- כללי תכן לשמירה על מסלול שעון ישיר | |
| 144 | ----- העדפת השימוש באפשרויות סינכרוני על פני טכניקות אפשרויות אחרות | |
| 145 | ----- איך ליצור מערכת שבה חלקים מסוימים פועלים בתדר נמוך יותר? | |
| 146 | ----- שלושת סוגי הכניסות ותפקידן | |
| 147 | ----- הימנעות מחוג משוב צירופי | |
| 149 | ----- הימנעות משימוש במונים א-סינכרוניים | |
| 150 | ----- הימנעות משימוש במחוללי פולסים א-סינכרוניים | |
| 152 | ----- אוסף תרגילי תכן | |

| עמוד | תת-נושא | פרק |
|------|---|----------|
| 163 | תכן גרפי היררכי עם Quartus | 5 |
| 164 | מבוא | |
| 166 | הסבר על המונה BCDUP וחלקיו | |
| 168 | יצירת המונה BCDUP | |
| 171 | ביצוע סימולציה למונה BCDUP | |
| 173 | יצירת Symbol למונה BCDUP | |
| 174 | הסבר על הדקי המונה BINDOWN ואופן פעולתו | |
| 176 | יצירת קובץ BINDOWN.BDF והיררכיה עליונה זמנית חדשה בשם BINDOWN | |
| 177 | יצירת המונה | |
| 182 | השלמת התכן ובדיקה של המונה BINDOWN.BDF | |
| 184 | יצירת רגיסטר היציאה REGOUT | |
| 185 | חיבור החלקים ויצירת היררכיה עליונה חדשה לפרוייקט | |
| 188 | דיבוג בסיסי של המערכת בחמרה | |
| 189 | מימוש סופי ובדיקה סופית של המערכת בחמרה | |
| 191 | הפעלה עצמית של המערכת | |
| 193 | רכיבי זיכרון ב - Quartus | 6 |
| 194 | מבוא לרכיבי זיכרון מסוג ROM | |
| 195 | הכרת ה - LPM_ROM | |
| 197 | יצירת אות סינשוואידלי | |
| 199 | יצירת טבלת ה - ROM | |
| 200 | הכרת הפורמט של קובץ MIF | |
| 202 | הכרת פורמט של קובץ HEX | |
| 204 | יצירת קובץ MIF לאות סינשוואידלי באמצעות היכולות של שפת VHDL | |
| 207 | יצירת טבלה לאות סינשוואידלי באמצעות MIFGEN.EXE | |
| 209 | הפיכת הקובץ sin256x8.mif לקובץ sin256x8.hex | |
| 210 | חיווט גרפי של המונה ורכיב ה - LPM_ROM | |
| 212 | סימולציה פונקציונלית של המחולל באמצעות Quartus | |
| 215 | תיאור טכסטואלי אלטרנטיבי של המערכת | |
| 216 | יצירת ממיר מבינארי ל - BCD | |
| 217 | תיאור טכסטואלי אלטרנטיבי של מערכת ההמרה bin2bcd | |
| 218 | רכיבי זיכרון כתיבה וקריאה | |
| 219 | בדיקת LPM_RAM_DQ בסימולציה | |
| 220 | הכרת רכיבי Dual Port RAM פשוט | |
| 222 | בדיקת ה - Dual Port RAM הפשוט | |
| 224 | שימוש ב - Dual Port RAM ליצירת Delay | |
| 226 | הכרת רכיבי FIFO | |
| 227 | בדיקת רכיב FIFO | |

| פרק | תת-נושא | עמוד |
|-----|---|------|
| 7 | תכנון בסיסי בשפת VHDL עם Modelsim ו Quatrus | 229 |
| | מבוא לתיאורים גרפיים וטכסטואליים | 230 |
| | אפשרויות שונות לביצוע מהלך התכנן – Design Flows | 233 |
| | סימולציה חיצונית ב - Modelsim לעומת סימולציה פנימית | 236 |
| | Quatrus | |
| | מבוא לתכנן מעורב (גרפי טכסטואלי) ושמירה על ביצוע | 237 |
| | Design-Flow מודרני | |
| | הכנות לקומפילציה של הגוזר הסינכרוני ב - Modelsim | 238 |
| | קומפילציה של קובץ הגוזר באמצעות ModelSim | 241 |
| | יצירת קובץ ה - Test-Bench לגוזר וקומפילציה שלו | 242 |
| | הרצת הסימולציה של הגוזר באופן ידני | 243 |
| | סימולציה באמצעות script של ModelSim | 245 |
| | סינתזה של הגוזר ב - Quartus | 246 |
| | סינתזה ובדיקת התכנן בחמרה | 249 |
| | הסתייעות בכלי עזר חשובים למעקב אחר הסינתזה של הקוד שלך | 250 |
| | ניתוח זמנים סטטי | 252 |
| | הכנת הקבצים לסימולציה לאחר הסינתזה באמצעות Quartus | 255 |
| | הכנת סביבת העבודה של Modelsim לסימולציה עם אטומים של | 258 |
| | Cyclone ורכיבים נוספים | |
| | ביצוע הסימולציה של הגוזר בפועל | 261 |
| | ביצוע שיפורים ב - Script התמיכה ברכיבי Altera | 264 |
| | תרגילי תכנן נוספים | 266 |
| 8 | דוגמאות ליצירת רכיבים פשוטים בשפת VHDL ויישומים פשוטים | 267 |
| | מבוא | 268 |
| | יצירת ממיר שימושי מקוד הקסדצימלי לתצוגות Seven Segment | 268 |
| | תצוגות Seven Segments על לוחות ה - DE2 | 269 |
| | קוד אפשרי בשפת VHDL | 271 |
| | יצירת שיפורים בקוד של הממיר | 272 |
| | יצירת פרויקט ריק - ללא קבצים עדיין | 273 |
| | הכנת התשתית להיררכיה גרפית ב - Quartus | 274 |
| | מימוש מערכת הבדיקה | 276 |
| | חיווט אלטרנטיבי בשפת VHDL | 280 |
| | תרגיל תכנן - שינוי העוצמה של התצוגה | 281 |
| | תיאור תווים באמצעות תצוגת Seven-Segment | 284 |
| | ממיר מ - ASCII ל - לתצוגת Seven-Segment | 286 |
| | דוגמה לתרגיל תכנן - מונה שמשנה כיוון | 288 |
| | דוגמה נוספת לתרגיל תכנן - מונה שיחס החלוקה שלו הולך וגדל | 290 |
| | בדיקת שני המונים האחרונים בחמרה | 293 |
| | מבוא לדוגמת יישום פשוטה : שעון זמן אמת | 294 |
| | מונה בסיס זמן | 295 |
| | מונה בסיס זמן בשפת - VHDL | 296 |
| | סימולציה של מונה בסיס הזמן | 299 |
| | בדיקת מונה בסיס הזמן בחמרה שעל גבי הלוח | 301 |
| | מונה BCD בעל יחס חלוקה 60 | 302 |
| | קוד אפשרי של counter60.vhd | 303 |
| | סימולציה של counter60 | 305 |
| | בדיקה של counter60 על החמרה של הלוח | 306 |

| עמוד | תת-נושא | פרק |
|------------|--|-----------|
| 308 | שתי צורות אפשריות לתצוגה של השעה בשעון | |
| 310 | מונה בעל יחס חלוקה 24 | |
| 311 | דוגמאות לקוד של counter24.vhd | |
| 313 | סימולציה של counter24 | |
| 314 | בדיקה של counter24 על החמרה של הלוח | |
| 316 | יצירה של RTC פשוט | |
| 320 | תרגיל תכן - שיפורים ל - RTC | |
| 320 | תרגיל תכן נוסף - Stopper | |
| 321 | יצירת רכיבים גנריים שימושיים | 9 |
| 322 | מבוא | |
| 323 | יצירת רכיב Pipeline | |
| 326 | בדיקת רכיב מסוג pipe בחמרה | |
| 327 | הכרת יחידת LPF | |
| 330 | בדיקת מסנן LPF בחמרה | |
| 331 | טיימר מסוג On-Delay | |
| 335 | סימולציה של הטיימר מסוג On-Delay | |
| 336 | טיימר מסוג Off-Delay | |
| 338 | סימולציה של הטיימר מסוג Off-Delay | |
| 340 | בדיקת שני הטיימרים ton ו toff בחמרה | |
| 341 | מימוש טיימר מחזורי | |
| 342 | סימולציה של הטיימר מסוג Repeater | |
| 344 | תרגיל תכנון - מימוש מתג (Key) עם פעולת Auto-Repeat | |
| 346 | תרגיל תכנון - מימוש מתג (Key) עם פעולת Auto-Repeat בעלת שתי מהירויות | |
| 347 | תרגיל תכנון - מימוש מתג (Key) עם פעולת Double-Click | |
| 351 | שימוש ב - LPF למניעת - ריטוטים ממגעים מכניים | |
| 354 | יצירת ממשק כללי למתגים מסוג KEY בלוח DE2 | |
| 355 | בדיקת יחידת הממשק למתג (כולל De-Bounce) על לוחות UP1/2 | |
| 357 | יצירת ממשק כללי למתגים בשפת VHDL עם פעולת De-Bounce כפרמטר | |
| 359 | מימוש ממיר מ - BCD לבינארי בשיטת ההכפלות ב - 10 | |
| 361 | מימוש ממיר בינארי ל - BCD בשיטת החלוקות ב - 10 | |
| 363 | מהלך הניסוי לבדיקת ההמרות | |
| 366 | המרה מבינארי ל - BCD בשיטה אחרת | |
| 367 | תרגילי תכן | |
| 369 | VHDL – גרפית ל - | 10 |
| 370 | מבוא | |
| 371 | מושגים בסיסיים במכונות מצבים | |
| 375 | הקצאת מצבים | |
| 380 | תיאור המערכת שתמומש | |
| 384 | תיאור מכונת המצבים של גלאי התנועה בשפת VHDL | |
| 386 | סינתזה וסימולציה של המכונה ב - Quartus | |
| 389 | סימולציה של המכונה ב - ModelSim | |
| 392 | יצירת קוד של המונה שמתאים ל - Quartus | |
| 394 | סימולציה של המונה ב - Quartus או ModelSim | |

| עמוד | תת-נושא | פרק |
|------------|--|-----------|
| 395 | היררכיה טכסטואלית ויתרונותיה | |
| 399 | חיווט גרפי לעומת טכסטואלי ושילובם | |
| 401 | החיווט גרפי של המערכת ב - Quartus | |
| 402 | הפיכת התיאור הגרפי לתיאור טכסטואלי בשפת VHDL | |
| 405 | יצירת חיווט טכסטואלי עם שמות בעלי משמעות | |
| 409 | סימולציה של כל המערכת | |
| 411 | סינתזה סופית ובדיקה על גבי לוח DE2 | |
| 412 | תרגילי תכן | |
| 413 | החטאה של פליפ-פלופים | 11 |
| 414 | מבוא | |
| 416 | תכן שאינו סובל מבעיות מסוג זה - תכן סינכרוני | |
| 418 | מניעת הגעה למצבים שגויים | |
| 421 | מניעת החטאה של אות כניסה | |
| 422 | אתחול מדורג של פליפ-פלופים | |
| 423 | הסבר על מבנה מערכת הניסוי - חלק א' (החטאות של din) | |
| 428 | מבנה מערכת הניסוי - חלק ב' (החטאות של resetN) | |
| 429 | מבנה מערכת הניסוי - חלק ג' (מניעת החטאות של din ו resetN) | |
| 430 | מהלך הניסוי - חלק ג' (מניעת החטאות של din ו resetN) | |
| 433 | הכרת כלי דיבוג בחמרה | 12 |
| 434 | מבוא | |
| 435 | נתח לוגי חיצוני ופנימי | |
| 437 | הכרת מושגים בסיסיים – מספר רמות המתח ותדר הדגימה | |
| 440 | הכרת מושגים בסיסיים : Trigger מבנה זיכרון הדגימה | |
| 441 | הכרת מושגים בסיסיים – הקלטה בזיכרון סיבובי לפני ואחרי הדרבון | |
| 442 | הכרת מושגים בסיסיים – הקלטה בזיכרון שמחולק לחלקים | |
| 443 | מבוא לשימוש ב - SignalTap | |
| 444 | עריכת תוכן של זיכרון וקבועים בחמרה בזמן אמת | |
| 445 | הסבר על מערכת הניסוי | |
| 447 | סינתזה של מערכת הניסוי | |
| 447 | סימולציה פנימית בסביבת Quartus לבדיקת המונה והגזור | |
| 448 | ביצוע סימולציה פונקציונלית חסרת זמנים לבדיקת גרטרור הסינוס | |
| 451 | הפיכת כל התכן לשפת VHDL | |
| 452 | התבוננות בקבצי הפרויקט ובדיקה מהן הספריות שלהן נזדקק | |
| 454 | בדיקת התמיכה של ModelSim שלך בספריות הנ"ל | |
| 455 | קומפילציה של הפרוייקט | |
| 456 | הפיכת קבצי MIF לקובצי HEX | |
| 459 | הורדת התכן לחמרה ובדיקתו | |
| 460 | קונפיגורציה בסיסית של ה - Signal TAP | |
| 466 | קונפיגורציה של שרשרת ה - JTAG בחלון ה - Signal-TAP | |
| 469 | שיוך הקובץ *.stp לפרוייקט וקומפילציה מחודשת של הפרוייקט | |
| 470 | צריבת הפרוייקט מתוך חלון ה - Signal-Tap | |
| 471 | הפעלת ה - Signal-Tap | |
| 474 | הפעלת ה - Signal-TAP במצבים של Auto-run ו Read Data | |

| פרק | תת-נושא | עמוד |
|-----------|---|------------------|
| | שינוי תצוגה של הסקלה האופקית ליחידות זמן אמיתי | 475 ----- |
| | שימוש ב - Trigger הכולל כמה תנאים | 478 ----- |
| | שימוש ב - Trigger סדרתי הכולל כמה רמות | 479 ----- |
| | התבוננות ביחידת הזיכרון sinrom | 481 ----- |
| | יצירת זיכרון sinrom שניתן לשינוי בזמן הריצה באמצעות Quartus | 483 ----- |
| | הפעלת ה - In-System Memory Content Editor | 487 ----- |
| | קונפיגורציה של שרשרת ה - JTAG בחלון ה - ISCME | 488 ----- |
| | צריבת הפרוייקט מתוך חלון ה - ISCME | 490 ----- |
| | קריאת תוכן הזיכרון | 491 ----- |
| | שינוי ידני של התוכן של הזיכרון | 492 ----- |
| | מילוי (Fill) של התוכן של הזיכרון | 493 ----- |
| | ביצוע Import או Export של קובץ MIF או HEX | 495 ----- |
| 13 | VGA תצוגת | 497 ----- |
| | מבוא לתצוגת VGA | 498 ----- |
| | תזמון אותות VGA | 502 ----- |
| | תצוגת Super-VGA | 506 ----- |
| | תדרי השעון בלוחות התרגול ושימוש בהם לתצוגת VGA | 507 ----- |
| | מחולל הסנכרון (Sync Generator) | 510 ----- |
| | תכן של מחולל הסנכרון - המונה האופקי | 514 ----- |
| | תכן של מחולל הסנכרון - המונה האנכי | 517 ----- |
| | תכן של מחולל הסנכרון - אותות אופציונליים | 519 ----- |
| | תכן של מחולל הסנכרון - דרגת סנכרון היציאות | 521 ----- |
| | כתיבה וסימולציה של מחולל הסנכרון | 523 ----- |
| | תדפיס המלא של הקובץ vgasync.vhd | 525 ----- |
| | בדיקת גנרטור הסנכרון (בלוחות DE1 ו DE2) | 528 ----- |
| | בדיקת גנרטור הסנכרון (בלוחות אחרים) | 529 ----- |
| | יחידת שיכפול הצבע | 530 ----- |
| | בדיקת יחידת הסנכרון ושכפול הצבע | 531 ----- |
| | תרגיל נוסף - מסך מחליף צבעים באופן עצמי | 534 ----- |
| | מתאם רזולוציית צבע | 534 ----- |
| | הקוד של מתאם רזולוציית הצבע | 535 ----- |
| | בדיקת מתאם רזולוציית הצבע | 537 ----- |
| | תרגיל נוסף - שליטה על צבע תצוגת המסך ברזולוציה גבוהה | 538 ----- |
| | צביעת שטחים על המסך | 539 ----- |
| | מימוש מערכת רמזורים - מבוא | 541 ----- |
| | מימוש יחידת בקר הרמזורים | 544 ----- |
| | בדיקת מערכת הרמזורים על גבי לוח התרגול | 549 ----- |
| | תרגיל תכן עם ריבוע גנרי הניתן למיקום על המסך | 550 ----- |
| | תצוגת טכסט אלפאנומרית | 551 ----- |
| | קובץ mif | 557 ----- |
| | הממדים של תצוגת הטכסט | 562 ----- |
| | שינוי המיקום ההתחלתי של תצוגת הטכסט | 562 ----- |
| | יצירת תנועה של הטכסט על המסך | 564 ----- |
| | יצירת תצוגת טכסט משתנה | 566 ----- |
| | תצוגת טכסט שמשתנה במהירות | 567 ----- |
| | יצירת תצוגה מהירה שמשתנה לכל היותר פעם אחת בכל Frame | 568 ----- |
| | הגדלת תדר השעון המכסימלי באמצעות טכניקה של PipeLine | 569 ----- |

| עמוד | תת-נושא | פרק |
|------------|---|-----------|
| 571 | ----- | |
| 572 | ----- | |
| 573 | ----- | |
| | איזון רוחבי של ה - PipeLine שינוי צורת התווים בזמן פעולת המערכת כתיבת תווים אלפאנומריים בקוד ASCII | |
| 575 | ----- | 14 |
| | שימוש במקלדת PS/2 | |
| 576 | ----- | |
| 578 | ----- | |
| 580 | ----- | |
| 581 | ----- | |
| 588 | ----- | |
| 590 | ----- | |
| 591 | ----- | |
| 595 | ----- | |
| 601 | ----- | |
| 602 | ----- | |
| 605 | ----- | |
| 608 | ----- | |
| 611 | ----- | |
| 612 | ----- | |
| 613 | ----- | |
| 614 | ----- | |
| 615 | ----- | |
| 616 | ----- | |
| 618 | ----- | |
| | מבוא ונתונים על ממשק המקלדת ברמת החמרה מבוא ונתונים על ממשק בין מקלדת PS/2 ו Host ברמת ה - bit תיאר חלקי המערכת תכן יחידת ה - BITREC מבוא לתשדורת של המקלדת ברמת ה - byte יצירת רכיב עזר לבדיקת התכן בתיאור מבני בדיקת המקלט הטורי בחמרה תכן יחידת ה - BYTEREC חיווט של המערכת הסופית לקליטה מהמקלדת הפיכת התיאור הגרפי לתיאור טכסטואלי בשפת VHDL יצירת חיווט טכסטואלי עם שמות בעלי משמעות סימולציה של הפרויקט באמצעות Modelsim סינתזה של יחידה הבדיקה הסופית בדיקת יחידת המקלט המלאה על גבי לוח התרגול הפיכת לחיצה ארוכה על מקש לפולס מתמשך תרגיל תכנון התייחסות מקובלת למקשי בקרה מיוחדים התייחסות מקובלת למקשי בקרה שמחליפים מצב תרגיל תכן | |
| 621 | ----- | 15 |
| | תקשורת טורית עם UART | |
| 622 | ----- | |
| 623 | ----- | |
| 625 | ----- | |
| 626 | ----- | |
| 632 | ----- | |
| 635 | ----- | |
| 637 | ----- | |
| 640 | ----- | |
| 645 | ----- | |
| 648 | ----- | |
| 650 | ----- | |
| 652 | ----- | |
| 654 | ----- | |
| 655 | ----- | |
| 656 | ----- | |
| 658 | ----- | |
| 660 | ----- | |
| | מבוא המשדר ומבנה התשדורת הטורית של UART עקרון הפעולה של המקלט אותות נוספים בתקן RS232 מידע חשמלי ופיסי נוסף על RS232 בדיקת המחשב והכבל המוצלב באמצעות LoopBack פעולות בסיסיות שתכניות Terminal מבצעות יצירת תקשורת טורית לבדיקת המערכת באמצעות התכנית COMSH יצירת תקשורת טורית לבדיקת המערכת באמצעות התכנית Tera-Term Pro יצירת תקשורת טורית לבדיקת המערכת באמצעות התכנית PuTTY יצירת תקשורת טורית עם HyperTerminal התבוננות באותות של ה - UART באמצעות SignalTAP תיאור ה - UART שיבנה תיאור הדקים שקשורים למשדר התיאור פנימי של המשדר בדיקת מערכת המשדר בסימולציה בדיקת מערכת המשדר בחמרה | |

| עמוד | תת-נושא | פרק |
|------------|---------|-----------------------|
| 662 | ----- | |
| 663 | ----- | |
| 665 | ----- | |
| 666 | ----- | |
| 668 | ----- | |
| 670 | ----- | |
| 671 | ----- | |
| 672 | ----- | |
| 675 | ----- | פרק הנספחים 16 |
| 676 | ----- | |
| 684 | ----- | |
| 689 | ----- | |
| 696 | ----- | |
| 708 | ----- | |
| 715 | ----- | |
| 718 | ----- | |
| 722 | ----- | |

פרק המבוא

מבוא לטכנולוגיה אלקטרונית של רכיבים מתוכנתים

שני "הקטרים" שמושכים כיום את הטכנולוגיה האלקטרונית קדימה הם ללא ספק: מעבדים (Microprocessors) ורכיבים מתוכנתים (Programmable Devices).

הטכנולוגיה של המעבדים, היא טכנולוגיה ותיקה יחסית, שהחלה בשנות השבעים וחוללה מהפכה בצורה שבה מממשים מערכות אלקטרוניות. מעבדים הם בעצם רכיבים בעלי מבנה פנימי קשיח. הגמישות ביישומים עם מעבדים, מושגת באמצעות כתיבת תכנה שונה לכל יישום. המפתח נעזר בדרך כלל במערכת פיתוח (Development System), על מנת לפתח את הפרויקט. לדוגמה המפתח עשוי להיעזר במערכת פיתוח של חברת Keil על מנת לפתח פרוייקט עם מעבד מסוג 89C51 או MPLAB כאשר הוא מפתח פרוייקט עם מעבד PIC. פרוייקטים שמבוססים על מעבדים מפותחים בעיקר בשפות C, ++C ובמידה מסוימת גם בשפת אסמבלי של המעבד.

הטכנולוגיה של רכיבים מתוכנתים היא אמנם צעירה יותר, אך כבר מזמן אינה טכנולוגיה אזוטריה שמיועדת רק ל"פריקים" של טכנולוגיה חדשנית. מדובר בטכנולוגיה מאוד בשלה שהיא בעלת השפעה עצומה על התעשייה, כבר למעלה משני עשורים. כיום כל איש אלקטרוניקה (מהנדס, הנדסאי וטכנאי) חייב להכיר טכנולוגיה חשובה זו. בניגוד למעבדים, שהם בעלי מבנה פנימי קשיח, לרכיבים מתוכנתים יש מבנה פנימי גמיש שאותו יוצר המתכנן באמצעות קונפיגורציה וחיווט של משאבי חמרה פנימיים ברכיב.

גם הטכנולוגיה של הרכיבים המתוכנתים חוללה מהפכה בצורה שבה מממשים מערכות אלקטרוניות ספרתיות. רכיבים מתוכנתים מתאימים ליישומים רבים, שבהם באה לידי ביטוי היצירתיות של המתכנן. המתכנן כבר אינו מוגבל בבחירת הרכיבים שלו לבחירה שנעשית מאוסף מצומצם של רכיבי מדף קיימים או לבחירה מאוסף מצומצם של יחידות חמרה שמצויות בתוך מעבד. המתכנן המודרני כבר מזמן אינו חייב להשתמש במספר גדול של רכיבי 74 ישנים על מעגל מודפס רחב ממדים, וזאת על מנת ליצור מערכת חמרה חדשה וייחודית. רכיבים מתוכנתים כוללים בתוכם כיום משאבי חמרה שהם שווי ערך למספר עצום של שערים (מיליוני שערים) ומספר עצום של פליפ-פלופים. רכיבים מתוכנתים מתאימים גם ליישומים מהירים שבהם מבנה המימוש מותאם ישירות לפעולת העיבוד המהירה הנדרשת. זהו מצב שונה מזה שקיים במעבדים שתוכננו להיות רכיבים אוניברסליים שעשויים להיות איטיים בביצוע של פעולת עיבוד ספציפית כל שהיא.

גם בפיתוח עם רכיבים מתוכנתים, המפתח נעזר במערכת פיתוח. הכלי - Quartus, שבו נעסוק בספר זה, מהווה דוגמה למערכת פיתוח שכזו. מדובר במערכת פיתוח של חברת Altera, שהיא כיום אחת מהחברות הדומיננטיות בשוק הרכיבים המתוכנתים. חברה דומיננטית נוספת בשוק הרכיבים המתוכנתים היא Xilinx. דוגמאות לחברות נוספות בשוק הרכיבים מתוכנתים הן למשל: Atmel, Lucent, Cypress, Lattice ועוד.

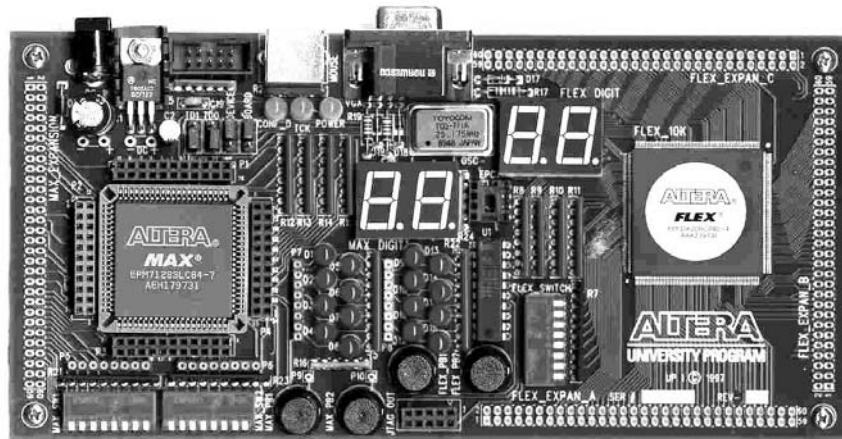
כשם שפרוייקטים שמבוססים על מעבדים מפותחים בארץ בעיקר בשפת C, פרויקטים שמפותחים עם רכיבים מתוכנתים מפותחים בארץ בדרך כלל בשפת תיאור חמרה VHDL. שפת VHDL היא גם שפת תיאור החמרה הנפוצה ביותר באקדמיה. שפות נוספות שבהן משתמשים לפיתוח עם רכיבים מתוכנתים הן: Verilog, AHDL ו ABEL. שפות VHDL ו Verilog הן שפות סימולציה כלליות. שתי השפות האחרונות (AHDL ו ABEL) הן שפות סינתזה של יצרני רכיבים ספציפיים (Altera ו Xilinx בהתאמה) והשימוש בהן פוחת במשך הזמן.

פיתוח של כרטיס אלקטרוני מודרני, משלב לעתים קרובות בין שתי הטכנולוגיות הנ"ל (מעבדים ורכיבים מתוכנתים) באותו הכרטיס. בשנים האחרונות, השילוב נעשה אף באותו הרכיב - כלומר בתוך הרכיב המתוכנת עצמו. דוגמאות למעבדים כאלו הם למשל: מעבד ה - NIOS של חברת Altera ו מעבד ה - MicroBlaze של חברת Xilinx. פתוח של פרוייקט משולב ברכיב מתוכנת, יכול לכלול למשל פעילויות כמו כתיבת קוד בשפת C, עבור מעבד שמריץ יישומים מורכבים תחת מערכת הפעלה (Real Time Operating System) - למשל תחת גרסה מיוחדת של Linux. מצד שני, פיתוח פרוייקט יכול לכלול למשל גם כתיבת קוד VHDL ליצירת החמרה של הרכיבים הפריפריאליים הייחודיים והמהירים שמקיפים את המעבד. לגישת תכן כזו קוראים: System On a Programmable Chip או בקצור - SOPC.

האבולוציה של כרטיסי תרגול לרכיבים מתוכנתים של Altera

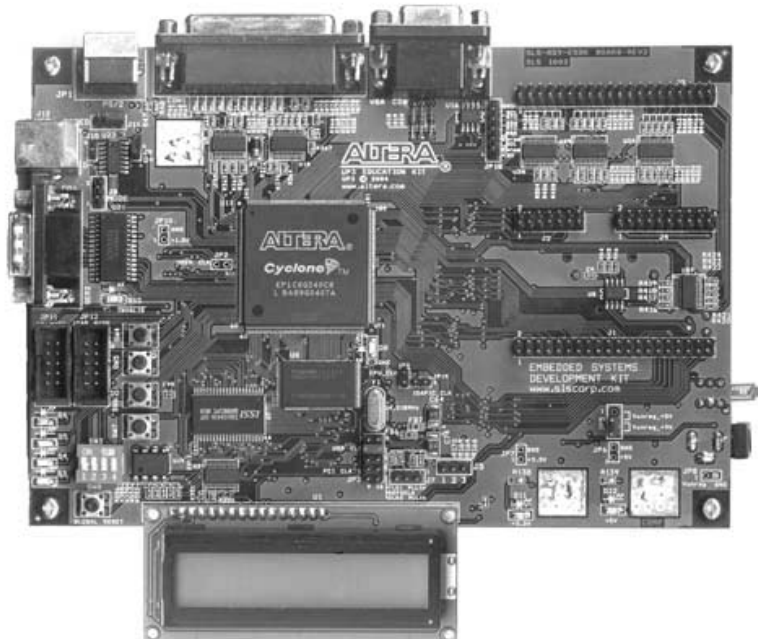
חברת Altera מצטיינת בתמיכה שהיא נותנת למוסדות לימוד. פרט לתמיכה הנדיבה במתן רשיונות למערכת הפיתוח Quartus, החברה תומכת ביצירת כרטיסי תרגול שונים עבור מכללות.

הלוח הראשון שנוצר למטרה זו היה ה- UP1 (המשמעות של האותיות היא University Program #1). לוח זה יוצר החל משנת 1999 והוא כלל שני רכיבים מתוכנתים: רכיב בלתי נדיף מסוג EPM7128SLC84 (נראה בצד שמאל באיור הבא) ממשפחת ה- MAX-7K ורכיב נדיף גדול יותר EPF10K20RC240 (נראה בצד ימין באיור הבא) ממשפחת ה- FLEX-10K.



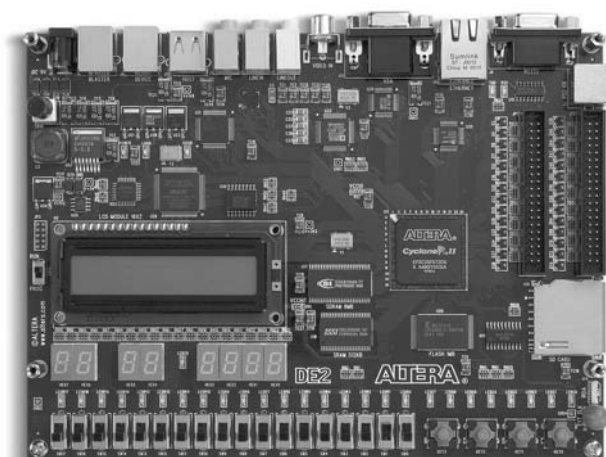
הכרטיס כלל גם: מספר קטן של לחצנים (שניים לכל רכיב), כמה מתגי DIP_SWITCH, תצוגות LED ו Seven-Segments, אופציה לחיבור הדקי חיבור סטנדרטיים החוצה, מחבר PS/2 ויצאת VGA (מאוד שימושי ונחמד) ומחולל גבישי בתדר של 25.175 MHz שמתאים במדויק לתצוגת VGA. בין השנים 2001 ל- 2002 הכרטיס הני"ל הוחלף בכרטיס UP2 שכלל רכיב FLEX חזק יותר שנקרא EPF10K70RC240. ייצורו של כרטיס זה נמשך עד למחצית של 2006.

ב- 2004 הוחל (על ידי חברת SLS) בייצור כרטיס תרגול חדש ומשוכלל יותר שנקרא UP3.



כרטיס זה כלל רכיב מתוכנת EP1C6Q240C8 ממשפחת ה- Cyclone-I. הלוח היה משופע ברכיבי עזר ואביזרים שונים כולל: חיבור למסך VGA, חיבור לתקשורת טורית כולל ממיר מתחים לתקן RS-232, חיבור PS/2, חיבור USB, חיבור IDE וחיבורים כלליים נוספים, חיבור לרכיבי I2C, חיבור לזיכרון FLASH, SRAM ו SDRAM, 4 לחצנים, 4 מתגי DIP-SWITCH, כמה נוריות LED, תצוגת LCD אלפאנומרית בעלת שתי שורות וכמה ומחוללי שעון בתדרים שונים (הרכיב המתוכנת כולל גם רכיבי PLL שיכולים לשנות את תדר השעון הפנימי).

בשנת - 2006 הוחל (על ידי חברת Terasic) ביצור לוח תרגול חדש ומאוד משוכלל בשם DE2.



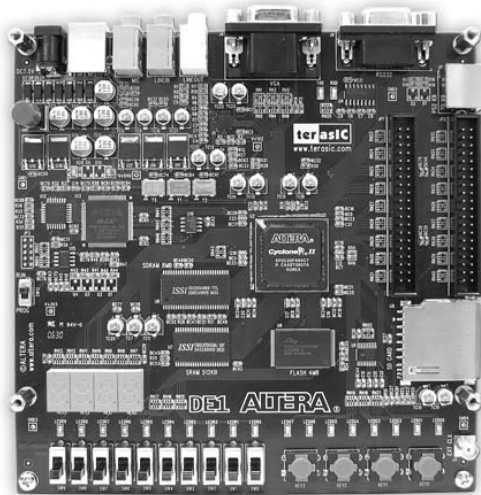
לוח זה כולל רכיב EP2C35F672 ממשפחת ה- Cyclone II. הלוח משופע מאוד ברכיבי עזר ואביזרים. בין השאר הלוח כולל:

- 18 מתגים
- 4 לחצנים עם De-Bounce
- 8 תצוגות Seven-Segment
- שמונה נוריות LED ירוקות
- 18 נוריות LED אדומות
- מחבר VGA עם רכיב DAC ברוחב 10 סיביות לשלושת הצבעים
- מחבר תקשורת טורית עם ממיר מתחים לתקן RS232
- משדר ומקלט אינפרא אדום תואם לתקן IrDA
- רכיב CODEC בן 24 סיביות עם כניסות ויציאות אנלוגיות נשלט I2C
- מפענח TV נשלט I2C
- מחבר PS/2 (שמיועד לעכבר או מקלדת)
- תצוגת LCD אלפאנומרית בעלת שתי שורות
- מחבר Ethernet עם רכיב ממשק
- רכיב זיכרון SRAM
- רכיב זיכרון SDRAM
- רכיב זיכרון FLASH
- שני מחברי הרחבה סטנדרטיים ברוחב 40 הדקים
- מחבר לכרטיס זיכרון SD-Card או MMC-Card
- שני מחברי USB
- מערכת צריבה באמצעות כבל USB או באמצעות רכיב זיכרון בלתי נדיף על הלוח
- כמה מחוללי שעון כולל אפשרות לשינוי התדר הפנימי על ידי PLL ברכיב המתוכנת

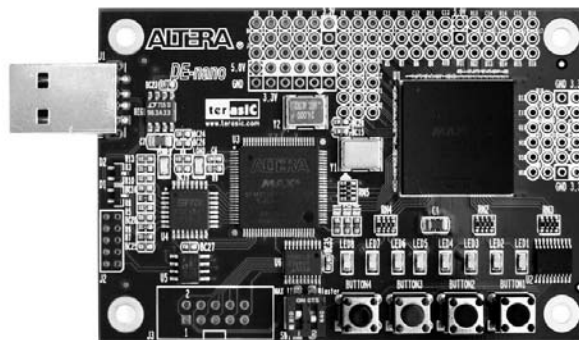
הרכיב המתוכנת שעל הכרטיס DE2, כולל משאבי חמרה רבים ומגוונים וכפי שניתן לראות, גם הכרטיס עצמו כולל מגוון גדול של רכיבים ואביזרים. על כרטיס זה ניתן להתלוצץ ולומר, שניתן לעשות אתו כל דבר פרט לקפה! הכרטיס DE2 יועד לביצוע ניסויים במעבדות במכללות ועבור פרויקטים (Development & Education Board). ספר זה מבוסס ברובו על כרטיס תרגול זה!

לפרויקטים גדולים יותר ניתן להשתמש בגרסאות של כרטיס DE2 שהן חזקות יותר כמו למשל: DE2-70, שכולל רכיב מתוכנת חזק יותר ומשאבי זיכרון ו Video חזקים יותר.

גרסה מוגבלת יותר של כרטיס ה- DE2, יוצרה כחצי שנה מאוחר יותר בשם DE1 והיא יועדה להיות גרסה זולה יותר שמיועדת לרכישה על ידי סטודנטים לתרגול עצמי בבית ולפרויקטים.



כרטיס זול נוסף שיוצר על ידי חברת Terasic הוא ה- DE-nano.



כרטיס זה כולל רכיב EPM2210F324 ממשפחת ה- MAX-II. מדובר ברכיב בלתי נדיף של Altera. הכרטיס כולל: ארבעה מתגים, שמונה נוריות LED בצבעים שונים, מחולל שעון בתדר 50 MHz ואופציה לחיבור הדקי חיבור סטנדרטיים החוצה. אספקת המתח והקונפיגורציה של הרכיב המתוכנת נעשים באמצעות כבל USB.

לקבלת מידע נוסף על כרטיסי התרגול האחרונים תוכל להיעזר באתרים הבאים:

<http://www.terasic.com.tw/>

<http://www.altera.com/>

ייעודו של הספר

הספר הוא ספר מעשי, שמטרתו לעזור לקורא להיכנס לעולם המופלא של התכן הספרתי המודרני, שנעשה באמצעות רכיבים מתוכנתים (CPLDs או FPGAs) ושפות תיאור חמרה (Hardware Description Language).

הספר מכיל תרגילי מעבדה רבים שמודגמים על לוח התרגול הבלתי רגיל DE2. למרות מה שנאמר על לוח התרגול הנ"ל, בכל זאת ניתן לבצע רבים מהניסויים שבספר גם על לוחות אחרים כמו: UP1, UP2, UP3, DE1, DE2-70, DE-nano ולוחות תרגול מסחריים שונים של המעבד NIOS.

כלי הפתוח שנבחר הוא כמובן Quartus, אך בנוסף אליו נעשה בספר שימוש בכלי הסימולציה החיצוני Modelsim של חברת Model-Technology או (Mentor). למרות שנעשתה בחירה בכלי הסימולציה החיצוני הזה, ניתן כמובן להשתמש גם בסימולטורים חיצוניים אחרים כמו Active-HDL או כל סימולטור חיצוני אחר. גם קוראים שאין ברשותם כלי לסימולציה חיצוני כל שהוא, יוכלו להפיק תועלת רבה מספר זה, מכיוון שהם יוכלו לבצע סימולציות פנימיות באמצעות Quartus.

שים לב, לכל הכלים הנ"ל קיימות גרסאות מוגבלות בזמן או גרסאות סטודנטים שניתן להורידן באינטרנט וניתן לקבל עבורן רשיון חוקי באמצעות האינטרנט. בנוסף לכך, מכללות יכולות לקבל גרסאות בלתי מוגבלות של תכנות אלו בחינם או במחיר מאוד מוזל.

בכדי להקל על הקורא ובכדי לא להגדיל יתר על המידה את ממדי הספר, נבחרה שפת תיאור חמרה אחת: שפת VHDL, שכפי שנאמר קודם, היא שפה נפוצה בארץ לתכן עם רכיבים מתוכנתים והנפוצה ביותר באקדמיה.

ללימוד שפת VHDL ניתן להסתייע בספר: "לימוד שפת VHDL לסימולציה וסינתזה" מאת עמוס זסלבסקי או בספרים אחרים. למרות שהשפה שנבחרה היא שפת VHDL, ניתן באופן עקרוני להמיר את הקודים בספר לשפות אחרות כמו AHDL או Verilog.

הקורא ייחשף בספר לנושאים מגוונים כמו:

- מהלכי תכן (Design Flows) שונים
- הכנסת תכן (Design Entry) גרפי וטכסטואלי ושילוב בניהם
- שימוש ברכיבים קשיחים וגמישים (עם פרמטרים)
- יצירת תכן היררכי באופנים שונים
- עקרונות תכן סינכרוני
- עבודה עם מונים, רגיסטרים ורכיבים חשבוניים ואחרים
- עבודה עם רכיבי זיכרון
- עבודה עם מכונות מצבים
- כתיבת קוד בשפת VHDL
- בצוע סימולציה לפני סינתזה
- קביעת הרכיב והקצאת הדקים שלו (Assignments & Settings)
- בצוע סינתזה והתבוננות בתוצאותיה
- אנליזת זמנים סטטית (Timing Analysis)
- טכניקות להגדלת תדר שעות
- ביצוע סימולציה אחרי הסינתזה
- הורדת התכן לרכיב המתוכנת
- שימוש בנתח לוגי (Logic Analyzer) שמוכנס לחמרה
- שינוי תוכן של רכיבי זיכרון וקבועים בחמרה בזמן העבודה
- החטאות של פליפ-פלופים ומניעתם

- התחברות לנוריות LED ותצוגות Seven Segments
- התחברות לתצוגת VGA ויצירת תצוגות גרפית וטכסטואלית שונות
- ממשק התחברות להתקן PS2
- ממשק התחברות ל – UART
- ועוד ...

הספר מתאים למתן ליווי מעשי לקורסים תיאורטיים בנושאים הבאים: "מערכות ספרתיות", "תכן לוגי" ו "שפת תיאור חמרה" שניתנים באוניברסיטאות ובמכללות. הספר יכול לשמש באופן ישיר גם כאוסף תרגילי מעבדה למעבדות ספרתיות ומעבדות לרכיבים מתוכנתים שניתנים באוניברסיטאות ובמכללות. סטודנטים שבמצעים פרויקטים בתחום יכולים להסתייע רבות בספר זה. הספר מתאים כמובן גם למהנדסי והנדסאי אלקטרוניקה בתעשייה שרוצים להכיר או להרחיב את ידיעותיהם בתחום חדשני זה בעצמם. כמובן שגם קוראים אלו יוכלו להשתמש בחלק מהדוגמאות והקודים בפרויקטים שהם יבצעו בתחום.

הספר יכול לשמש גם כהקדמה לקורסים בנושאי מבנה מעבדים ומעבדים שמוכנסים לרכיב מתוכנת (System On a Programmable Chip) כדוגמת ה – NIOS או מעבדים אחרים.

מבנה הספר והמלצות ללומד

הגישה המקובלת בקורסים רבים במכללות ובאוניברסיטאות, היא לחשוף את הלומד לנושא של רכיבים מתוכנתים **תחילה באמצעות תיאורים גרפיים**. תיאורים אלו נחשבים לידידותיים יותר (בהשוואה לתיאור בשפות תיאור חמרה), היות והם דומים לתיאורים שאותם הכיר הלומד בקורסים עיוניים כמו "מערכות ספרתיות". בדרך כלל **בהמשך, הלומד נחשף באמצעות מעבדות או קורסים נוספים לשפת VHDL** והמשך התרגול עם רכיבים מתוכנתים נעשה באמצעות שפה זו. בשלב זה הלומד נחשף כמובן גם ליתרונותיה של צורת תיאור זו.

ששת הפרקים הראשונים של הספר, מבוססים ברובם על תכן גרפי והם ניתנים ללימוד על ידי קוראים שהם עדיין חסרי רקע מוקדם בשפת VHDL. כמובן שפרקים אלו יכולים להילמד גם לפני או במקביל ללימוד קורס בשפת VHDL וכהכנה לחלקים הבאים של הספר, שדורשים תשתית ידע בשפת VHDL. כפי שנאמר כבר קודם, ליצירת תשתית ידע בשפת VHDL, כדאי להסתייע בספר לימוד כל שהוא (למשל "לימוד שפת VHDL לסימולציה וסינתזה" מאת המחבר של ספר זה). חלק מהנושאים שמופיעים בפרקים מתקדמים יותר, יכולים גם הם להתבצע באופן גרפי שמתאים לקוראים מהסוג זה (חלקים מסוימים מפרקים 8, 11, 12 ו 13).

בששת הפרקים הראשונים שבספר, נשתמש בסימולטור הפנימי של Quartus. שתי הסיבות לכך הן: הקלה על הלומד וחשיפתו לסימולטור גרפי פשוט וידידותי וההנחה שהקורא ממילא עדיין אינו מיומן מספיק בשפת VHDL ואינו מסוגל לכתוב Test-Bench.

שאר הפרקים בספר (החל מפרק 7), מבוססים גם על תיאורים בשפת VHDL והם מתאימים לקוראים שיש להם כבר ידע בשפת VHDL או שנמצאים בתהליך לימוד של השפה. קוראים שנמצאים בשלב לימוד השפה, יכולים בעצם להשתמש בשני הספרים: ספר זה, והספר "לימוד שפת VHDL לסימולציה וסינתזה" **ביחידות לימוד משותפות**. כמובן שקוראים שיש להם ידע בשפת VHDL יכולים להפיק תועלת מכל חלקיו של ספר זה, כולל גם מהפרקים הראשונים שבספר, שנותנים לקוראים אלו רקע טוב יותר בהכרת רכיבים ספרתיים, הכרת יסודות תכן ספרתי סינכרוני והכרת תהליך הפיתוח עם Quartus.

בפרקים המתקדמים יותר של הספר (החל מפרק 7) ניתן להשתמש הן בסימולטור הפנימי של Quartus (למי שאין ברשותו Modelsim) והן בסימולטור החיצוני של Modelsim. הסימולציות ב – Modelsim נעשות בעיקר ברמת Source Level, אך הן יכולות להיעשות גם ברמת Gate Level (כלומר לאחר הסינתזה).

למי שאכן נמצא ברשותו סימולטור חיצוני כמו Modelsim והוא מיומן בכתיבת Test Bench בשפת VHDL, עדיף לבצע בשלב זה את כל הסימולציות באמצעות כלי זה.

נאמר שבחלקים המתקדמים של הספר, נעשה שימוש בתיאורים בשפת VHDL. האם התיאורים יהיו טכסטואליים בלבד או מעורבים ? נושא זה דורש הבהרה והוא יוסבר כאן.

אנשים רבים ששולטים היטב בשפות תיאור חמרה, מעדיפים לעתים לתאר את החמרה באופן טכסטואלי. אנשים אלו יכתבו כמובן את התיאורים ההתנהגותיים שלהם בשפת VHDL, ובנוסף לכך גם את כל התיאורים המבניים (החיווטים) שלהם בשפת VHDL. לעומתם, אחרים בכל זאת יעדיפו להשתמש בשילוב בין טכסט וגרפיקה. תיאורי הטכסט ישמשו אותם לתיאור התנהגויות והתיאורים הגרפיים ישמשו אותם לתיאור המבנה של המערכת. כותב ספר זה, שייך לאסכולה הראשונה של האנשים, כלומר לאלו שמעדיפים לכתוב הכל בטכסט ! למרות מה שנאמר כרגע על גישתו של המחבר, חלקו המתקדם יותר של ספר זה (החלק שמתחיל מפרק 7), נכתב ברגישות ונעשה מתוך התחשבות גם באנשים שמעדיפים "לראות" חיווטים שמתוארים באופן גרפי. כמובן שאין מניעה מלבצע את כל התרגילים בחלק המתקדם של הספר כאשר החיווטים הגרפיים מוחלפים בחיווטים בשפת VHDL, כלומר כל אחת משתי צורות העבודה היא אפשרית.

אחת הבעיות שעלולה להתעורר **לכאורה**, כאשר מתארים מערכת באופן מעורב (באמצעות גרפיקה וטכסט) היא, שלא ניתן לבצע לכל המערכת סימולציה מהירה בשפת VHDL **לפני** שנעשית סינתזה כל שהיא. היכולת לבצע סימולציה לפני הסינתזה, היא חשובה כאשר מבצעים תכן של מערכות גדולות ומשתמשים במהלך תכן (Design Flow) מודרני שדורש לבצע סימולציה לפני סינתזה.

למעשה, בעיה כזו אינה קיימת כאשר משתמשים ב - Quartus. כלי זה מסוגל להפוך שרטוט (תיאור מבני גרפי) לרשימת חיווט בשפת VHDL. הפיכת ממוחשבת של חיווטים גרפיים ל - VHDL, פותחת בפנינו אלטרנטיבה נוחה ליהנות משני העולמות: כלומר יצירה של חיווטים מורכבים באופן גרפי וידידותי, וזאת מבלי שיש צורך לוותר על הדרישה של Design Flow המודרני, שמחייב לבצע סימולציה לפני סינתזה. מה שאנו בעצם עושים הוא להשתמש ב - Quartus ככלי ל - Design Entry. בתעשייה קיימים כלי תכנה (צד שלישי) רבי עוצמה אחרים שנועדו בדיוק למטרה זו. דוגמה לכלי כזה הוא למשל HDL-Designer של Mentor-Graphics.

להלן רשימת הפרקים של הספר

1. הכרת Quartus ותכן גרפי בסיסי
2. הכרת מונים ב - Quartus
3. הכרת רגיסטרים ב - Quartus
4. תכן סינכרוני
5. תכן גרפי היררכי עם Quartus
6. רכיבי זיכרון ב - Quartus
7. תכן בסיסי בשפת VHDL עם Modelsim ו Quartus
8. דוגמאות ליצירת רכיבים פשוטים בשפת VHDL ויישומים פשוטים
9. יצירת רכיבים גנריים שימושיים
10. מימוש עם מכונת מצבים והפיכת היררכיה גרפית ל - VHDL
11. החטאה של פלי-פלופים
12. הכרת כלי דיבוג בחמרה
13. תצוגת VGA
14. שימוש במקלדת PS/2
15. תקשורת טורית עם UART

נתאר בקצרה את פרקי הספר השונים.

פרק הראשון תבצע הכרות ראשונה עם כלי הפיתוח - Quartus. תיאור התכן יעשה באופן גרפי. בניסוי זה תשתמש ברכיבים פשוטים שנקראים primitives (שכוללים שערים ופליפ-פלופים) ובמהלך הפרק תתנסה במימוש מערכות צירופיות ומערכות סינכרוניות פשוטות. בנוסף לכך תלמד כיצד לבצע בחירה של הרכיב והדקיו. במהלך הפרק תלמד גם כיצד לבצע סימולציה ברמת Gate-Level עם הסימולטור הפנימי של Quartus. במהלך הפרק תלמד גם כיצד לבצע הורדה של התכן ללוח התרגול.

הפרק השני והשלישי עוסקים באבני הבניה הסינכרוניים החשובים: מונים ורגיסטרים. פרקים אלו עוסק באופן ספציפי בהכרת מונים ורגיסטרים של Quartus. תחילה יוצגו בקצרה מונים ורגיסטרים קשיחים ששייכים למשפחת ה-74 המיתולוגית. בהמשך תכיר רכיבים גמישים ושימושיים יותר שנקראים בטרמינולוגיה של Quartus בשם Megafuctions. התצורה של הרכיב הגמיש שכוללת את ההדקים והפרמטרים הנבחרים של הרכיב, יכולה להיקבע באופן ישיר או באמצעות שימוש בכלי ידיותי שנקרא MegaWizard. שתי צורות הקביעה הללו מתורגלות בפרקים אלו.

הפרק הרביעי עוסק בתכן סינכרוני. בתחילתו של הפרק מוסבר מהו תכן סינכרוני, מהם יתרונותיו, כיצד מחשבים את ביצועי התזמון שלו וכיצד ניתן לשרם. בהמשכו של הפרק מוצגת סדרה של כמה כללי "עשה" ו"אל תעשה" מאוד חשובים שינתבו אותך לביצוע תכן סינכרוני נקי. בפרק זה נסביר גם מה עלול לקרות כאשר לא מקפידים על כללים אלו ומהן הודעות האזהרה ש-Quartus עלול לשגר. בסופו של הפרק מצוי אוסף של תרגילי תכן פשוטים שנעשה באמצעות: מונים, רגיסטרים ומערכות צירופיות נוספות. ביצוע המטלות שבתרגילים אלו צריך להיעשות תוך כדי הקפדה על הכללים שנלמדים בפרק זה, כלומר עליך להקפיד לבצע תכן סינכרוני נקי!

הפרק החמישי מתאר כיצד יוצרים תכן גרפי היררכי. השימוש בהיררכיה שימושי וחשוב כאשר מתארים מערכות גדולות ומורכבות. ניתן לקבוע בכל שלב בתכן, היררכיה עליונה זמנית של הפרוייקט וניתן לסנתז ולסמלץ אותה.

הפרק השישי מתמקד ברכיבי זיכרון ב-Quartus. בארכיטקטורות הרכיבים של Altera קיימים משאבי זיכרון פנימיים. בכדי לנצל משאבים אלו, אנו נדרשים להשתמש ברכיבי זיכרון שאותם מכיר הכלי Quartus. בפרק זה נשתמש ברכיבי זיכרון מסוג Dual Port-RAM, RAM, ROM ו-FIFO.

הפרק השביעי הוא הפרק הראשון שבו תיצור בעצמך רכיבים חדשים פשוטים שלא קיימים במאגר הרכיבים של Quartus, באמצעות תיאורם בשפת -VHDL. בפרק זה נסתפק בדוגמה צנועה של גוזרים סינכרוניים, אך מכאן ואילך הספר ישתמש באופן נרחב בטכניקה של כתיבה טכסטואלית. בהשוואה לתיאורים גרפיים - לכתיבה טכסטואלית יש יתרונות רבים. בפרק זה ובפרקים שבהמשך, תיחשף ליתרונות אלו באופן מעשי. עם בשלב זה חסר לך עדיין ידע בשפת VHDL, כאן אולי המקום שבו יהיה עליך להרחיבו (למשל באמצעות קריאת פרקים 2 עד 5 ואולי גם 6 בספר: "לימוד שפת VHDL לסימולציה וסינתזה").

הפרק השמיני ניצור כמה רכיבים שימושיים נוספים ונחברם ביחד, על מנת ליצור יישומים פשוטים. מדובר למשל ברכיבים שמפעילים תצוגות Seven Segment ומונים מסוגים שונים. היישומים הפשוטים שניצור יהיו למשל RTC (Real Time Clock), שמציג את ספירת השניות, הדקות והשעות ומאפשר כיוון שלו, ושעון עצר (Stopper). בחלק מהרכיבים השימושיים שתיצור בפרק זה תוכל להשתמש גם בהמשכו של הספר ובפרוייקטים. את התיאור של הרכיבים ניצור בשפת VHDL. את הקוד של הרכיבים הפשוטים שניצור נוכל לקמפל ולסמלץ באמצעות Modelsim (או Quartus) ואת החיבור של הרכיבים נבצע באופן גרפי או באופן טכסטואלי. בפרק זה תוכל גם להתנסות בסימולציה ברמת Gate Level של Quartus ותיווכח שהיא מאוד איטית ותוכל להשוותה לסימולציה מאוד מהירה שנעשית באמצעות Modelsim.

פרק התשיעי בספר, תיצור רכיבים שימושיים נוספים, אך הם לא יהיו רכיבים קשיחים, אלא רכיבים גמישים בעלי פרמטרים. בפרק זה תיצור רכיבים שימושיים שונים שבהם תוכל להשתמש בהמשכו של הספר ובפרויקטים. מדובר ברכיבים כגון: רכיבי Pipeline, טיימרים מסוג Off-Delay, On-Delay, Repeater, מסנן מעביר נמוכים ספרתי (LPF) וממירים בין ייצוגים בינאריים ו BCD. בפרק זה ניצור גם יישום מאוד שימושי לפרויקטים: ממשק נוח וגמיש למתג מכני. לשם כך נשתמש בארבעה רכיבים מהרשימה הנ"ל ובגוזר (שאותו יצרנו בפרק 7). ממשק זה יכלול בין השאר גם את האופציה של מניעת ריטוטים של מגעים מכניים (De-Bounce), הפעלה בלחיצה או בעזיבה, פעולת Auto-repeat (כפי שקיימת במקלדות של PC), ופעולת Double-Click (כפי שקיימת במקשי העכבר).

היכולת לשלוט על פרמטרים בשפת VHDL, מבוססת על פסוקי generic map, ו generic map ועל תשתיות נוספות שמכוסות בעיקר בפרק 8 ובמידה מסוימת גם בפרק 10 בספר: "לימוד שפת VHDL לסימולציה וסינתזה". אם אינך מכיר תשתית חשובה זו עכשיו הזמן להכירה.

פרק 10 עוסק בתיאור של מערכת שכוללת גם מכונת מצבים. בחלק זה תבצע בנוסף לסימולציה של כל החלקים של המערכת גם סימולציה כללית של כל המערכת ב - VHDL. לפני שתבצע פעולה זו, יהיה עליך להפוך באמצעות Quartus את התיאור הגרפי של המערכת ל - VHDL.

פרק 11 עוסק בתופעה של החטאה של פליפ-פלופים. תופעה זו נגרמת מאי קיום זמני הכנה, החזקה או זמן שחרור מפעולה א-סינכרונית בפליפ-פלופים. בניסוי זה תתבונן בתופעה ותנסה למנוע אותה בכרטיס התרגול.

פרק 12 עוסק בהכרה של שני כלי עזר רבי עצמה. אחד הכלים הוא נתח לוגי (Logic Analyzer) שמוכנס לתוך (Embedded Logic Analyzer) ונקרא בטרמינולוגיה של Quartus בשם SignalTAP. כלי שני שאותו תכיר בפרק זה, מאפשר לשנות או לקרוא תוכן של רכיבי זיכרון וקבועים בחמרה, תוך כדי פעולתו של הרכיב המתוכנת. כלי זה נקרא בטרמינולוגיה של Quartus בשם: In-System Content Memory Editor.

פרק 13 עוסק בתצוגת VGA. בתחילתו של הפרק תיצור מחולל סנכרון (VGA Sync Generator). לאחר מכן הפרק עוסק בצביעת משטחים בצבעים שונים על המסך ובכתיבת טכסט. מסך ה - VGA הוא מסך נוח מאוד לשימוש עם לוחות תרגול של Altera והוא מאפשר להציג אותות שונים בנוחות רבה. פרק זה עשוי להיות שימושי מאוד בפרויקטים.

פרק 14 עוסק ביצירת מערכת ממשק בסיסית למקלדת PS/2. בהמשך תוכל להשתמש במקלדת לפרויקטים שונים ותוכל אף לשכלל בעצמך את הממשק כך שהוא יהיה גם דו כיווני ויכול להשתמש גם בעכבר. גם פרק זה עשוי להיות שימושי מאוד בפרויקטים.

פרק 15 הוא הפרק הלפני אחרון בספר והוא עוסק ביצירת רכיב UART בסיסי. רכיב זה יאפשר לך לחבר את לוח התרגול למחשב PC ולהעביר אותות בין לוח התרגול והמחשב בנוחות. גם פרק עשוי להיות שימושי מאוד בפרויקטים.

פרק הנספחים, כולל בשלושת החלקים הראשונים שלו, רקע תיאורטי קצר על ארכיטקטורות של רכיבים מתוכנתים. לאנשים שהם חסרי ידע בתחום זה, מומלץ לקרוא חלקים אלו. לקבלת מידע מפורט יותר על ארכיטקטורות של רכיבים מתוכנתים תוכל להיעזר בספרות שעוסקת במקצוע "מערכות ספרתיות" ובדפי יצרן של Altera. המחבר של ספר זה עוסק בימים אלו בכתיבת של ספר מקיף בנושא "מערכות ספרתיות" שיצא לאור בקרוב בהוצאת ספרים זו ושיכלול רקע תיאורטי רחב על ארכיטקטורות של רכיבים מתוכנתים.

פרק הנספחים כולל גם תדפיסים של סקריפטים של לוחות התרגול השונים.

כיצד פותחו החומרים שבספר

ספר זה מבוסס על התנסות רבה של המחבר בהעברת קורסים בנושא רכיבים מתוכנתים. הקורסים הועברו באתרים שונים ברחבי הארץ ועבור קהלי יעד שונים.

חלק מתכניי הקורס נוצר מהרצאות, שלוו בשקפים, וחלק משקפים אלו שוכתב לחוברות לעבודה עצמית במעבדה. חלק נוסף מתכניי הקורס נוצר באופן ישיר כתרגילי מעבדה.

מקורם של חלק גדול מהתכנים בספר (עד פרק 12) הוא מניסויי מעבדה אינטנסיביים שכתבתי עבור סטודנטים להנדסאים בטכניון במעבדה הספרתית (6.6) ובמעבדה לרכיבים מתוכנתים (6.27) מקצוע חובה חדש בכל המכללות). מקורם של חלק גדול מהתכנים המתקדמים יותר בספר (פרקים 13 ואילך) הוא מניסויי מעבדה ומהדרכה של קבוצות פרויקט גמר רבות של בוגרי אלקטרוניקה המעולים של מכללה זו. הכנת כל חוברות הניסויים הללו אילצה אותי לנסח את ההסברים בצורה מאוד ברורה והשאלות הרבות של הסטודנטים ובעיקר במהלך הפרוייקטים שהנחתי, סיפקו לי משוב חשוב בקשר לאיכות ההסברים.

מאוחר יותר חלק קטן מהתכנים של חוברות הניסוי הנ"ל (להנדסאים) שונה והותאם גם לצורכי סטודנטים להנדסה שבפקולטה להנדסת חשמל בטכניון. הניסויים שנוצרו עבור סטודנטים אלו מתבצעים בשישה מפגשים בסימסטר החמישי או השישי. מדובר במעבדות חובה (מעבדה 1 לסטודנטים בהנדסת חשמל - 044151, ומעבדה 1 לסטודנטים במסלול להנדסת מחשבים - 044160).

חלק מהתכנים (בעיקר אלו שבתחילת הספר ושאינם כוללים שימוש בשפת VHDL), הותאמו גם לצורכיהם של סטודנטים להנדסת חשמל ואלקטרוניקה וסטודנטים להנדסת מחשבים במכללות אקדמיות שונות שמתנסים במעבדה ספרתית בסיסית שניתנת לפני קורס מלא בשפת VHDL.

חלק מהתכנים (כמו פרקים 1, 5 ו 13 ותכנים רבים נוספים בספר) הוכן גם עבור קורסים של Altera שנעשים בתיאום עם חברת איסטרוניקס במכללת הייטק. מדובר בקורס Altera למתחילים (קורס מספר - 516 במכללת הייטק וקורסים דומים במכללת מדיאטק בעלי המספרים 513/516). בגרסאות הישנות של תכנים אלו, עשיתי גם שימוש רב בעבר בקורסים שונים שהדרכתי בחברות שונות בתעשייה (למשל ברפא"ל, באלביט ובחברות אחרות).

כמוכן שחלק מהתכנים בספר כוללים גם אלמנטים שונים שבהם השתמשתי בעצמי בקודים שפיתחתי בפרוייקטים שונים עבור חברות שונות בתעשייה. בפרוייקטים אלו בדרך כלל השתמשתי בגרסאות מורכבות יותר מאלו שמוצגות בספר זה. בספר ניסיתי לעשות שימוש בעיקר באלמנטים שהם מאוד פשוטים וקלים להבנה.

הרבה מהתכנים המקוריים שבספר נכתבו בעבר הרחוק (לפני שנים) ובהם נעשה שימוש בכלי הפתוח המיתולוגי - Max+Plus II והם יועדו במקור ללוחות UP1 ו UP2. כל התכנים הישנים הללו עודכנו כמוכן והפיתוח בספר הנוכחי מתבסס כולו על Quartus והתרגילים שוכתבו והם מיועדים בעיקר להרצה על לוח DE2, שהוא הלוח הנפוץ ביותר במכללות וגם בתעשייה נוהגים להשתמש בו. גרסאות ה - Quartus העכשוויות הן מאוד בשלות ויציבות מבחינת המבנה של ה - GUI שלהן ולכן אין חשיבות רבה לגרסת Quartus הספציפית שבה משתמשים. התרגילים במהדורה זו הורצו על גרסאות 6 עד 9. למרות מה שנאמר כאן, כמוכן שגרסאות חדשות של Quartus הן ללא ספק טובות יותר מבחינת היכולות שלהן וכדאי תמיד להשתמש בחדשה ביותר.

בגרסאות ישנות של הניסויים נעשה שימוש בטכסט בכמה שפות: AHDL ו VHDL ולעתים גם ב - Verilog. צורת הכתיבה הכוללת ריבוי שפות, הכבידה מאוד על הקוראים של הגרסאות הישנות של הניסויים והפכה את מלאכת העדכון של הניסויים למסורבלת. בספר זה בוטלו כפילויות אלו ונעשתה בחירה של שפה אחת - VHDL וכפי שנאמר כבר קודם מדובר בשפה הפופולריות ביותר לפיתוח עם רכיבים מתוכנתים בארץ ובשפה הנפוצה ביותר באקדמיה.

תודות

ברצוני להודות למשתתפים בקורסים שקיימתי בנושאי רכיבים מתוכנתים בחברות שונות: ברפא"ל-לשם, ברפא"ל-מכון-דוד בתדיראן קשר, בתדיראן מערכות, RAD, ECI, אלביט, טלרד ועוד.

ברצוני להודות לאנשים הבאים שקראו חלקים שונים מהטכסט והעירו הערות מועילות: סידרוב אולג, פבל אפרוס, עמית הרשקוביץ, אנטולי איסטומין, סער מזרחי, ויקטור לויט, אמירה שהלא, אמיר שהלא, רקפת רוזנטל, הילה טאויל, ענבל אואסקר, טל יחיאל, שאדי פראג², עתליה זיו, רוני ממון, רז לזר, סבטלנה גנטישציק, מורן עזרן, יעלה אברהם, מיכאל מרגוליס, טניה מרגוליס, עדי הורן, עמית שוסטיק, הדר מורנו, אוריין אדרי, ברנדוין טל, שמולביץ נועה, בת-אל אלקובי, עדי זוארץ, שרון מרום, פבל קוצ'רז'וק, לאוניד סמרטנקו, יורי ספיבק, בידס עלא ותאמר, יסמין קדארי, שירה ברזילי, רועי ז'נו, כפיר שוויקי, נייש מיכאל ועודד שחם. תודה מיוחדת על ההערות הבונות גם לידידי סקלי ערן.

תודה נוספת מגיעה לאנשים המצויינים בחברת איסטרוניקס, שהיא הנציגה של חברת Altera בארץ. כמשתמש ותיק ברכיבים ובכלי התכנה של חברת Altera ויותר מאוחר גם כחבר ACAP (תכנית יועצים מומחים של Altera), התנסיתי בהעברת קורסים והדרכות שונות מטעמם (קורסי Altera: במכללת הי-טק ומכללת מדיאטק, אלביט חיפה, רפא"ל, ECI ועוד...). בנוסף לכך מגיעה להם תודה גדולה על התמיכה המעולה שהם נותנים למוסדות לימוד. ברצוני להודות במיוחד לאבנר אוזן, משה מועלם, עופרה אברהמי, ותמר לוסטגרטרן.

בנוסף לכך, מגיעות תודות לאנשים המצויינים בחברת סיטל (עופר הופמן וניר חמצני), שהם הנציגים של Modelsim וכלי תכנה נוספים של Mentor בארץ. במיוחד מגיעה להם תודה גדולה על התמיכה שהם נותנים למוסדות לימוד.

תודה מיוחדת מגיעה גם לשני אנשים מצויינים ונפלאים, אלי שושן ואינה ריבקין, מהמעבדה הספרתית בפקולטה להנדסת חשמל בטכניון שיזמו וליוו את תהליך ההמרה והתאמה של חלק מהתכנים שבספר זה ובספר VHDL שכתבתי, גם לניסויי סטודנטים להנדסה, שמתבצעים בסביבה הטכנולוגית ואנושית המורכבת של הפקולטה להנדסת חשמל בטכניון.

תודה מיוחדת מגיעה לשלושה אנשים נוספים שדרבנו ועודדו אותי לכתוב את הספר: יאן לרון, שהוא חבר טוב וכותב ספרים פורה בפני עצמו (שהוציא לאחרונה גם ספר בלתי רגיל בנושא SPICE...), שי מלול שעמו התייעצתי לעתים בזמן כתיבת הספר (ושעומד להוציא בקרוב שני ספרים מעולים בנושא מיקרו-בקרים 8951) ואריה ליבנה שהצביע ללא לאות על הצורך בכתיבה של ספר מעשי ורציני בעברית בנושא הרכיבים המתוכנתים.

תודה גם למוציא לאור: אלי מיטב על האמון וגם על הסבלנות.

אני מאחל לכל הקוראים קריאה מועילה ולא פחות חשוב מכך - הנאה וסיפוק בהפנמה של הטכנולוגיה המרתקת של רכיבים מתוכנתים שאותה חווים רבים בעיקר בזמן פתוח פרויקטים.

הגרסאות הישנות של רוב הניסויים עברו דיבוג אינטנסיבי על ידי מספר רב של קוראים. חלק מהגרסאות החדשות של הניסויים עבר גם הוא דיבוג על ידי קוראים נוספים. למרות כל המאמצים שלי בכתובת הספר ולמרות המשוב המועיל הרב של האנשים המצוינים שהעירו לי הערות ותקנו לי שגיאות בגרסאות המוקדמות של הספר (כלומר בחוברות הניסוי הישנות), עדיין עלולות להיות בספר זה טעויות והן כמובן כולן באחריותי הבלעדית וכמובן איני חולק אחריות זו עם אחרים.

אודה לכל מי שישלח לי הערות בקשר לטעויות שכאלה או לכל מי שיצביע על בעיות כלשהן בהבנה של הטכסט או יציע לי הערות בונות כל שהן. כל הערה קטנה כגדולה, תתקבל על ידי בברכה. ליצירת קשר ניתן להשתמש בכתובות הדואר האלקטרוניות הבאות:

amos.books@gmail.com

amos.zaslavsky@gmail.com

או שניתן להתקשר לטלפונים הבאים:

050-7270673

04-8230219 (בערב)

תודה לאשתי וילדי (אפרת, טל ואורי) שסבלו בתקופת כתיבת ספר זה מהיעדרויות שלי מבילויים משפחתיים.

הספר מוקדש לזכרם של שני אנשים יקרים: שולמית ורפאל בר-אלי ז"ל.