



# IC2 - BUS Protocol Overview

## אלון צ'רני

מסמך זה הורד מהאתר <http://www.underwar.co.il>  
מחבר המסמך איננו אחראי לכל נזק, ישיר או עקיף, שיגרם עקב השימוש במידע המופיע במסמך, וכן לנכונות התוכן של הנושאים המופיעים במסמך. עם זאת, המחבר עשה את מירב המאמצים כדי לספק את המידע המדויק והמלא ביותר.

הודעה: אני מקווה כי לא נפלו טעויות במהלך הבנת הפרוטוקול, ובכל מקרה התקן הוא המחייב. (צ'רני אלון)  
המטרה של מסמך זה הינה לתאר את פרוטוקול התקשורת הסדרתית  $I^2C$  בצורה ברורה ומובנת ככל שניתן וזאת בניגוד לתקן עצמו שכתוב ב"שפה משפטית" ומטרתו להיות מדויק גם על חשבון הבהירות.

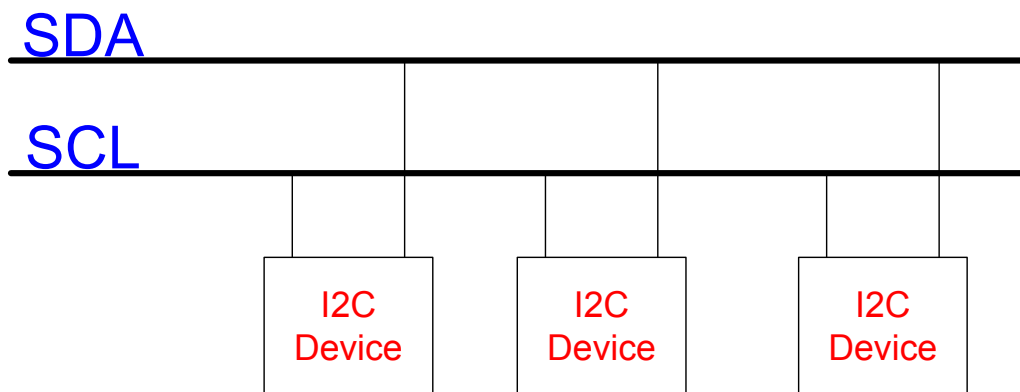
# I<sup>2</sup>C – PROTOCOL

## תוכן העניינים:

3	.....	רקע כללי	1
4	.....	עקרונות פעולה כלליים	2
5	.....	עקרונות פעולה	3
6	.....	Acknowledge	4
6	.....	סנכרון - מספר Masters על קו ה- SCL	5
7	.....	בוררות - מספר Masters על קו ה- SDA	6
8	.....	התהליך המלא	7
8	.....	סוגים שונים של תקשורת	8
9	.....	כתובת של 7 ביט. (או מה בא אחרי START)	9
10	.....	'קריאה כללית'	10
12	.....	חיבור מיקרו-קונטרולר ל- I2C-BUS	11
13	.....	נספחים	12

## 1 רקע כללי

- תקן שנקבע ע"י חברת Philips Semiconductors בתחילת שנות השמונים.
- $I^2C = Inter\ Integrated\ Curcuit\ Bus$
- התקן נועד לאפשר תקשורת טורית בין במחשב (Master) ובין מספר Slaves. המטרה של הפרוטוקול היא לספק "הסכם מסגרת", כלומר, הפרוטוקול מסדיר את אופן התקשורת בין שני משתמשים (Master and Slave). הפרוטוקול לא מתייחס ל-DATA אלא רק מגדיר מתי ואיך יש לשדר את ה-DATA. (סיבית אחת בכל מחזור שעון, 8 סיביות מידע בין שני אישורים). מאפשר אפילו למספר Master לתקשר עם מספר Slaves.
- יתרונות : פשוט לתפעול - 2 קווי תקשורת.
- קצבי העברת המידע :
  - עד ל- 100 Kbit/sec לפי ה- Standard Mode.
  - עד ל- 400 Kbit/sec לפי ה- Fast Mode.
  - עד ל- 3.4 Mbit/sec לפי ה- High-Speed Mode.
- התקן מאפשר זיהוי התנגשויות (CD), סנכרון שעון וביצוע Hand-shake.



- להלן השוואה בין הקצבים של הפרוטוקולים השונים של התקשורת הסדרתית.

## Speed of various connectivity methods (bits/sec)

CAN (1 Wire)	33 kHz (typ)
I <sup>2</sup> C ('Industrial', and SMBus)	100 kHz
SPI	110 kHz (original speed)
CAN (fault tolerant)	125 kHz
I <sup>2</sup> C	400 kHz
CAN (high speed)	1 MHz
I <sup>2</sup> C 'High Speed mode'	3.4 MHz
USB (1.1)	1.5 MHz or 12 MHz
SCSI (parallel bus)	40 MHz
Fast SCSI	8-80 MHz
Ultra SCSI-3	18-160 MHz
Firewire / IEEE1394	400 MHz
Hi-Speed USB (2.0)	480 MHz

## 2 עקרונות פעולה כלליים

- הממשק מורכב משני קווים דו-כיווניים : SDA ו-SCL.
- לכל רכיב יש כתובת ייחודית. כמו-כן כל רכיב יכול לתפקד במקלט או כמשדר.
- SDA - Serial Data line. SCK - Serial Clock line.
- ה-Master מוגדר כמי שיוזם את תהליך ההתקשרות עם רכיב מסוים שהוא ה-Slave.
- במהלך ההתקשרות גם ה-Master וגם ה-Slave משמשים הן כמשדר והן כמקלט.
- השעון מיוצר ע"י ה-Master בלבד, אך ה-Slave יכול להחזיק את השעון למטה בשביל חיווי.
- מספר הרכיבים המחוברים ל-Master תלוי בקיבול. הקיבול המכסימלי הינו 400 pF.
- לרכיב ממוצע יש קיבול של 10 pF.
- לפרוטוקול I<sup>2</sup>C יש 127 כתובות.
- I<sup>2</sup>C משתמש (ע"פ הגדרה) במתח של 5 וולט. (ניתן לשנות את המתח)
- יש תדר מכסימלי אך אין תדר מינימלי.
- ניתן לממש 'Plug & Play' בעזרת I<sup>2</sup>C וזאת באמצעות תוכנה האחראית על מיקרו שבודק האם רכיב מסוים מחובר או שיש לדלג על קטע הקוד הרלוונטי.

### 3 עקרונות פעולה

- ניתן לשנות את קו המידע SDA רק כאשר קו השעון SCL נמצא בנמוך.
- המידע על ה- SDA צריך להיות יציב בזמן שקו השעון SCL נמצא בגבוה.

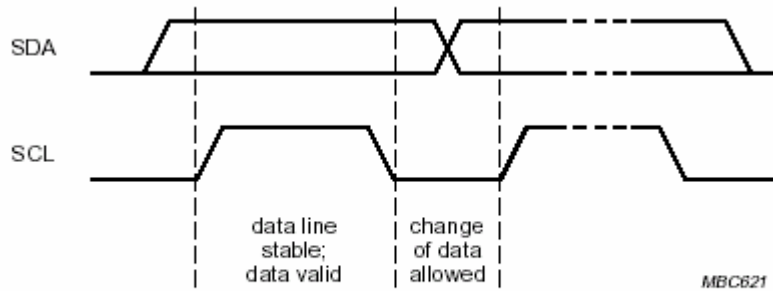


Fig.4 Bit transfer on the I<sup>2</sup>C-bus.

- אם יש שינוי על קו המידע SDA בזמן שקו השעון SCL גבוה אז :  
מעבר מ- HIGH ל- LOW פירושו START  
מעבר מ- LOW ל- HIGH פירושו STOP

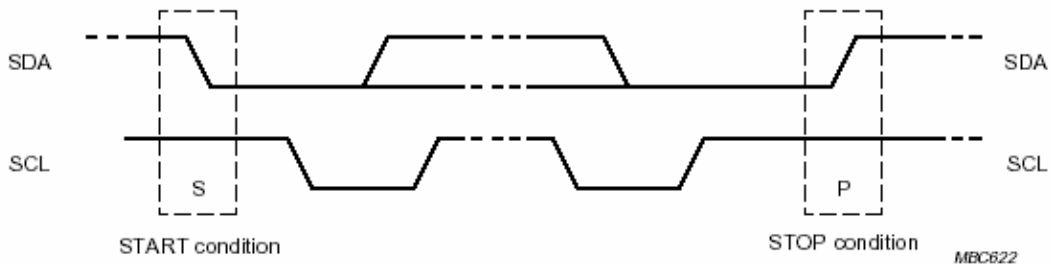


Fig.5 START and STOP conditions.

- ניתן לשים על קו המידע SDA, STOP במקום repeated START.
- אורך המילה על ה- SDA הוא בהכרח 8 ביט. אין הגבלה על מספר המילים שאפשר לשלוח.
- ה- MSB נשלח בתחילת המילה.
- ה- Slave יכול להוריד את ה- SCL, ל- LOW, ובכך ליצור מצב של WAIT ל- Master.

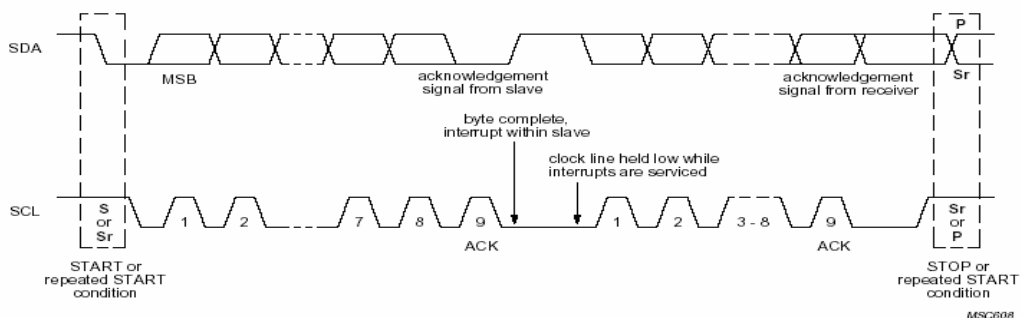


Fig.6 Data transfer on the I<sup>2</sup>C-bus.

## Acknowledge 4

- כל מילה חייבת להסתיים בביט תשיעי של Acknowledgement (Ack).
- השעון של ה-Ack מיוצר ע"י ה-Master.
- בזמן Ack המשדר שם מתח-צף על ה-SDA, והמקלט שם LOW על ה-SDA.
- כאשר ה-Slave לא משדר Ack על כתובתו (כי הוא לא יכול, למשל) הוא שם מתח-צף על ה-SDA בשביל לשחרר את הקו ל-Master אשר בוחר עם לחדש את ההתקשרות עם הרכיב הנוכחי אם לא.
- כאשר ה-Slave משדר Ack על כתובתו, אך שם-לב במהלך המשך ההתקשרות כי הוא לא יכול להמשיך לקלוט מידע. אז הוא משדר Not-Ack בביט התשיעי של המילה הבאה. אח"כ הוא שם מתח-צף על ה-SDA בשביל לשחרר את הקו ל-Master אשר בוחר עם לחדש את ההתקשרות עם הרכיב הנוכחי אם לא.
- אם המשדר של ה-Master מעורב בתהליך ההתקשרות, אזי בסוף המידע ה-Master לא שולח Ack למשדר של ה-Slave בביט התשיעי של המילה הבאה. ה-Slave שם מתח-צף על ה-SDA בשביל לשחרר את הקו ל-Master אשר בוחר עם לחדש את ההתקשרות עם הרכיב הנוכחי אם לא.

## סנכרון - מספר Masters על קו ה-SCL 5

- כל Master מייצר את השעון שלו בשביל ה-SCL.
- המידע valid רק כאשר SCL ב-HIGH.
- כאשר יש מספר שעונים הסנכרון יעשה בעזרת Wired-AND על ה-SCL. שעון עם זמן LOW הארוך ביותר קובע את אורך ה-LOW של ה-SCL. שעון עם זמן HIGH הקצר ביותר קובע את אורך ה-HIGH של ה-SCL. כלומר : SCL מקבל HIGH רק כאשר כל השעונים ב-HIGH.

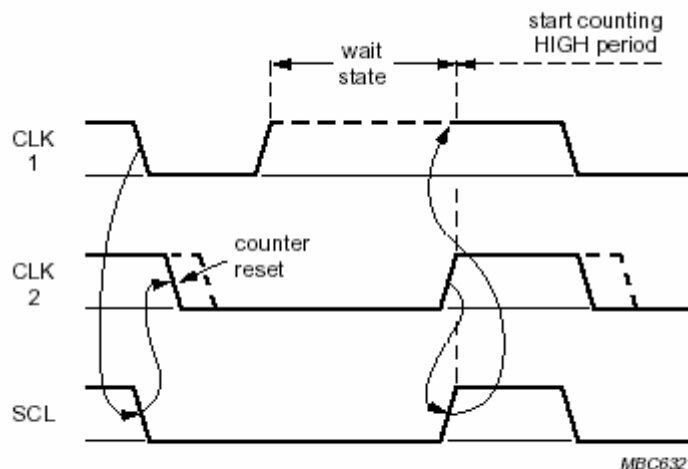


Fig.8 Clock synchronization during the arbitration procedure.

## 6 בוררות - מספר Masters על קו ה- SDA

- Master יכול לשדר רק כאשר ה- BUS פנוי.
- תהליך הבוררות מופעל כאשר יש שני Masters או יותר. אין עדיפות בין Masters שונים. ראשית, כל Master מנסה להתחיל את ההתקשרות, בזמן שהשעון ב- SCL ב- HIGH.
- ה- Master שנמצא ב- LOW בזמן ששאר ה- Masters נמצאים ב- HIGH מנצח.
- Master אשר רואה (המתבוננות ב- SDA) כי הפסיד "מכבה את עצמו".
- אם שני Masters מנסים להגיע לאותו רכיב, כלומר, נותנים את אותה הכתובת, הבוררות תעשה : על המידע אם ה- Master הוא משדר, או על ה- Ack אם ה- Master הוא מקלט.
- אין איבוד מידע במהלך הליך הבוררות. (המידע הוא של ה- Master המנצח)
- אסור לבצע תהליך בוררות :
- 1. בין Repeated START ובין המידע.
- 2. בין Repeated START ובין STOP..
- 3. בין STOP ובין המידע.
- באופן עקרונית Master אחד יכול "לתפוס" את ה- SDA לנצח.
- Slaves לא מעורבים בתהליך הבוררות.

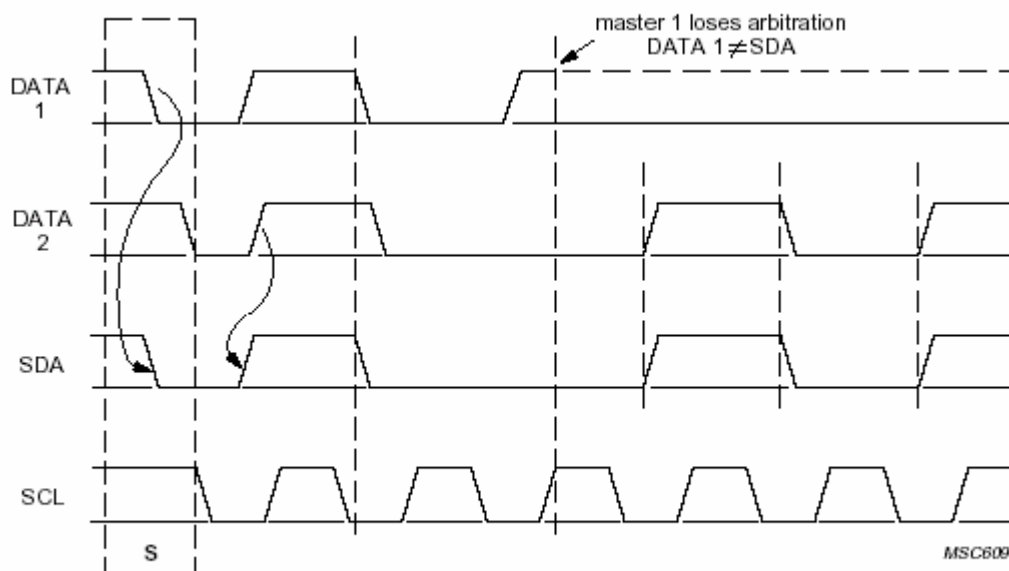


Fig.9 Arbitration procedure of two masters.

## 7 התהליך המלא

- ראשית על ה-Master להמתין ולבדוק כי ה-Bus פנוי, במידה וכן ניתן ליזום שידור.
- התהליך המלא דורש שה-Master ישלח כתובת של רכיב - 7 ביט. ועוד ביט שמיני של R/W
- $R/W = '0' \Rightarrow \text{WRITE}$        $R/W = '1' \Rightarrow \text{READ}$
- שליחת מידע מסתיימת ע"י STOP ע"י ה-Master.
- אם ה-Master רוצה להמשיך לתקשר על ה-BUS הוא מייצר Repeated START ואז הוא יכול לפנות לכל רכיב שירצה בלי STOP. כך מתאפשר תהליך הכולל גם קריאה וגם כתיבה בין STOP ל-START.

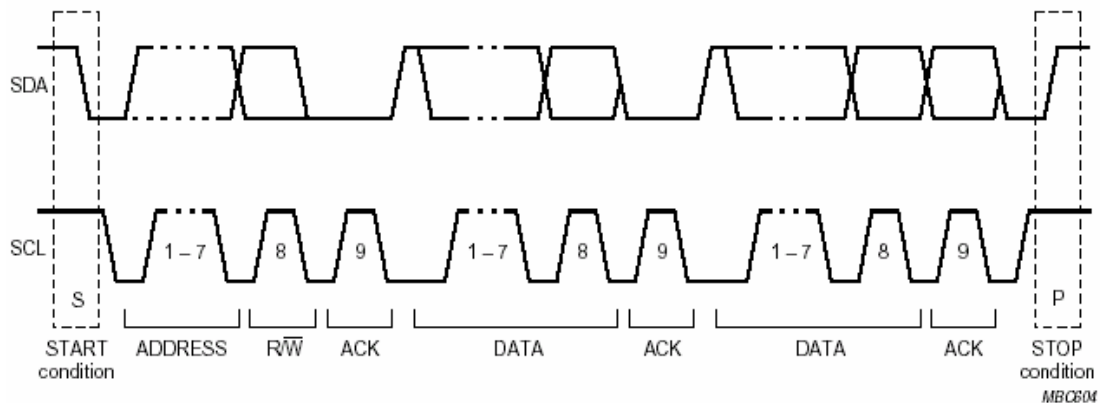


Fig.10 A complete data transfer.

## 8 סוגים שונים של תקשורת

1. ה-Master הוא המשדר לאורך כל ההתקשרות והוא כותב ל-Slave.

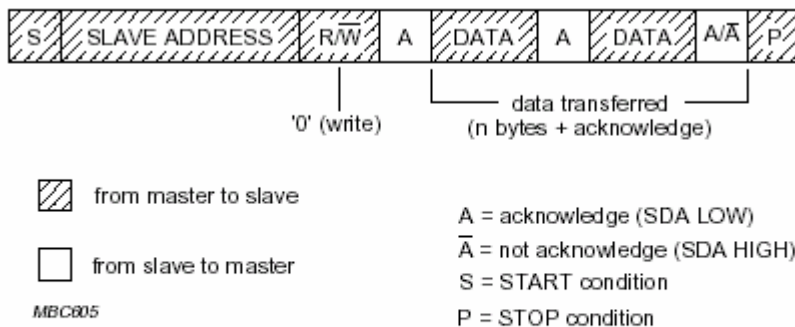


Fig.11 A master-transmitter addressing a slave receiver with a 7-bit address. The transfer direction is not changed.

2. ה- Master יוזם את ההתקשרות בשביל לקרוא מה- Slave.

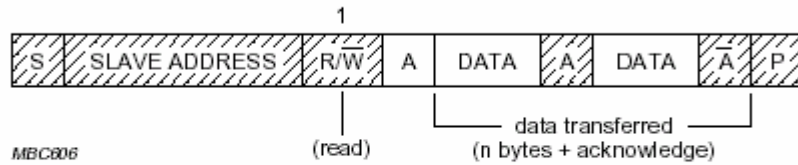


Fig.12 A master reads a slave immediately after the first byte.

3. ה- Master יוזם את ההתקשרות בשביל לקרוא ולכתוב ל- Slave. (שילוב של 1 ו-2)

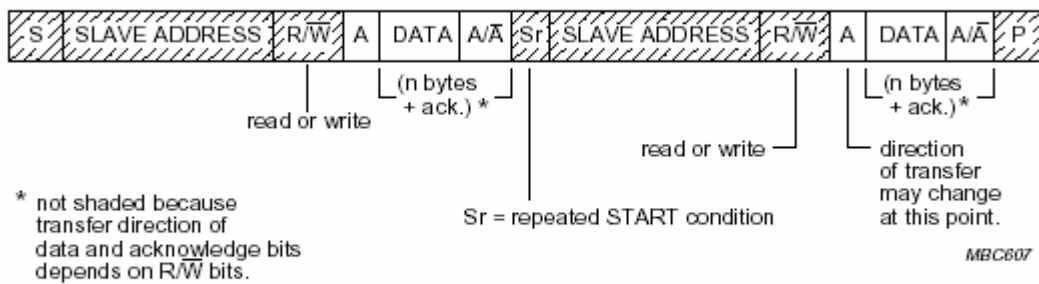


Fig.13 Combined format.

- הודעה ריקה אינה חוקית. (הודעה המורכבת מ- START ומיד אח"כ STOP)
- אחרי כל מילה יש Ack או NOT-Ack.

## 9 כתובת של 7 ביט. (או מה בא אחרי START)

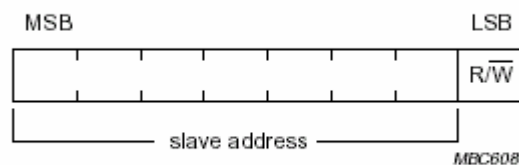


Fig.14 The first byte after the START procedure.

- עקרונית אחרי START תבוא כתובת של אחד הרכיבים. (Slaves) בפועל קיימת אפשרות של 'קריאה כללית' כאשר ה- Master קורא לכל הרכיבים להזדהות. ניתן לתכנן את הרכיבים כך שהם יתעלמו מה-'קריאה הכללית'.
- בדר"כ 'קריאה כללית' מורכבת משני מילים. מילה אחת משמשת כ-'קריאה כללית' והמילה השנייה משמשת כתיאור של הפעולה שיש לנקוט.
- ניתן להגדיר כי חלק מהכתובת קבוע וחלק ניתן לתכנות.
- להלן דוגמאות לכתובות שמורות :

**Table 2** Definition of bits in the first byte

SLAVE ADDRESS	R/W BIT	DESCRIPTION
0000 000	0	General call address
0000 000	1	START byte <sup>(1)</sup>
0000 001	X	CBUS address <sup>(2)</sup>
0000 010	X	Reserved for different bus format <sup>(3)</sup>
0000 011	X	Reserved for future purposes
0000 1XX	X	Hs-mode master code
1111 1XX	X	Reserved for future purposes
1111 0XX	X	10-bit slave addressing

## 10 'קריאה כללית'

- רכיב יכול להתעלם מ-'קריאה כללית' בכך שהוא לא שולח Ack אם הוא לא צריך את המידע.
- אם רכיב החליט כי הוא צריך את המידע הוא צריך לשלוח Ack אחרי ה-'קריאה הכללית' ואחרי כל המילים הבאות.
- אם רכיב לא יכול להתמודד עם חלק מהמידע הוא יכול להתעלם אך הוא צריך לשלוח NOT-Ack.
- המשמעות של 'קריאה כללית' נמצאת במילה השנייה.
- נתייחס לסיבית B שהינה ה- LSB בכתובת של המילה השנייה.

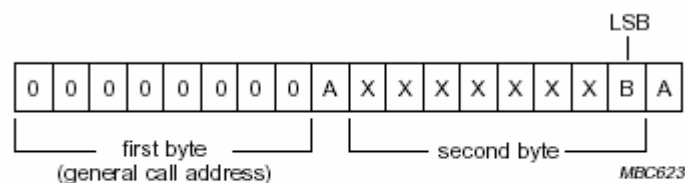


Fig.15 General call address format.

כאשר  $B = '0'$  נתייחס ל- 2 ערכים אפשריים של כתובות של המילה השנייה.

הבעיה: לפעמים ה- Master שולח קריאה ל- Slave מסוים, ואף אחד לא עונה.

הבעיה נובעת כאשר יש שני רכיבים עם אותו כתובת עצמית או אם עקב בעיות אתחול בתוך

הרכיב עצמו הכתובת העצמית של הרכיב לא עלתה כראוי.

הפתרון: שכל הרכיבים יכתבו את הכתובות העצמיות שלהם מחדש.

1. כתובת של  $(H'06') 00000110$

המשמעות: אתחול של כל הרכיב וכתובה מחדש של הכתובת העצמית של ה- Slaves.

2. כתובת של  $(H'04') 00000100$

המשמעות: כתיבה מחדש רק של הכתובת העצמית של ה- Slaves, ללא אתחול כל הרכיב.

- כתובת של  $(H'00') 00000000$  אסורה בשימוש במילה השנייה.

- שאר הכתובות לא מוגדרות והרכיבים אמורים להתעלם מהם.

- תהליכי כתיבת הזכרון מוגדרים בדפי הנתונים של הרכיבים.

כאשר  $B = '1'$  - 'קריאת חומרה כללית'

- ה- Master רואה שהוא לא מצליח לתקשר ולכן הוא מבצע 'קריאת חומרה כללית' הכוללת כתובת

של 'קריאה כללית' בכתובת המילה הראשונה ואת הכתובת שלו בכתובת של המילה השנייה.

כמובן, אף Slave לא מזהה את הכתובת כשלו. (שהרי הכתובת היא של ה- Master)

את הקריאה אמור לזהות 'רכיב חכם' דוגמת מיקרו-קונטרולר אשר יפנה הלאה את המידע.

(אם ה- Master הוא גם Slave הוא צריך לשים את כתובתו כ- Slave)

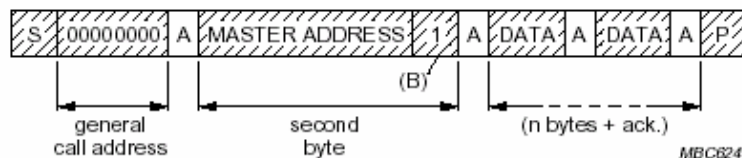


Fig.16 Data transfer from a hardware master-transmitter.

- אפשרות נוספת הינה להפוך את ה- Master כמסדר ל- Slave כמקלט.

כאשר ה- Master נמצא במצב זה (מקלט כ- Slave) מערכת קביעת תצורה של ה- Master

(A system configuring Master) יכולה לומר ל- Master את כתובת היעד של המידע.

לאחר סיום התהליך, ה- Master חוזר לתפקד כ- Master ומסדר.

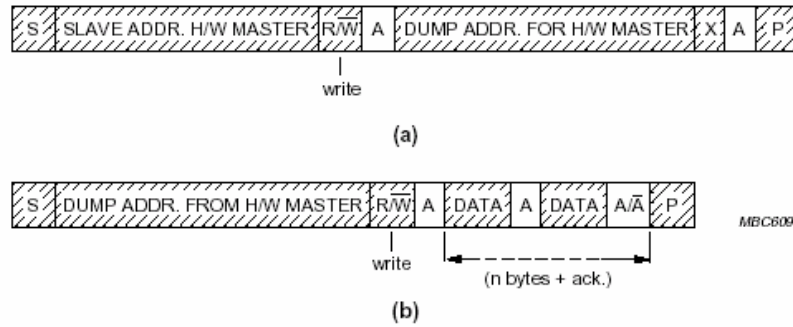


Fig.17 Data transfer by a hardware-transmitter capable of dumping data directly to slave devices.  
 (a) Configuring master sends dump address to hardware master  
 (b) Hardware master dumps data to selected slave.

## 11 חיבור מיקרו-קונטרולר ל- I2C-BUS

ישנם שני דרכים לחיבור מיקרו-קונטרולר ל-  $I^2C - BUS$  :

1. חיבור המיקרו-קונטרולר דרך ממשק לרכיב חומרה של  $I^2C$ . כך שדגימת ה-BUS נעשית באמצעות החומרה.
  2. חיבור המיקרו-קונטרולר דרך תוכנה. כך שדגימת ה-BUS נעשית באמצעות התוכנה.
- כמובן, דגימה באמצעות חומרה מהירה הרבה יותר. וככל שהדגימה מהירה יותר כך ההשהיה של העברה המידע קצרה.
  - כדי לאפשר למיקרו-קונטרולר גם דגימה איטית של ה-BUS נייצר START ארוך שיכלול :

1. תנאי של START
2. מילת START (00000001)
3. מחזור שעון של Ack
4. תנאי של Repeated Start

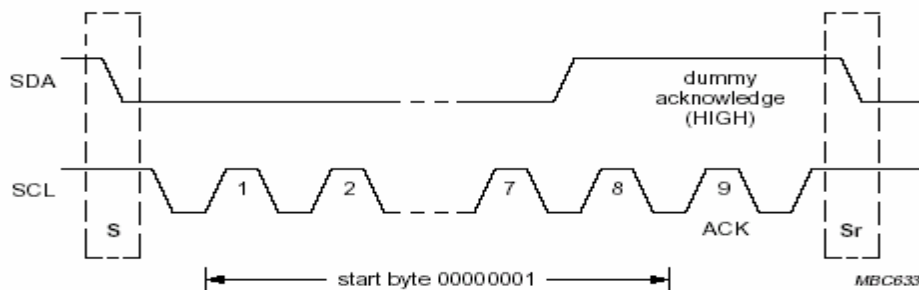


Fig.18 START byte procedure.

- כעת מיקרו-קונטרולר נוסף יכול לדגום את ה- SDA בדגימה איטית עד שהוא מזהה שה- SDA נמצא ב- LOW. כאשר הוא מזהה זאת הוא עובר לדגימה מהירה בשביל לזהות את התנאי של Repeated Start לצורכי סנכרון.
- כאשר מקלט מזהה את התנאי של Repeated Start הוא מתעלם ממילת ההתחלה START.
- לאף רכיב אסור לייצר Ack לאחר מילת ההתחלה START.

## 12 נספחים

עד כאן התקן הרגיל, מכאן ואילך התקן מפרט תוספות המתייחסות למספר מקרים :

1. Fast Mode - מאפשר קצב של עד 400 Kbit/sec
2. High-speed Mode - מאפשר קצב של עד 3.4 Mbit/sec
3. כתובת של 10-bit – מאפשר עד 1024 Slaves.