I2C Request and Response Encoding

1722b I2C Rewrite
May 16, 2023
Brian Edem

Existing I2C Encoding

Table 27 ~~—I2C ACF~~ Controller Request and Target Response field values

Request or Response	I2C_msg _payload	wr	akv	ack	rdv	c2t	rd	str	stp
Controller									
CR1-Start	<addr>	1	0		1	1	1	1	0
CR2-AC	<addr>	1	0		1	1	1	0	0
CR3-WC	<data>	1	0		0	1	0	0	0
CR4-WE		0	0		0	1	0	0	1
CR5-WR	<addr>	1	0		1	1	1	1	0
CR6-RC		0	1	1	0	1	1	0	0
CR7-RE		0	1	0	0	1	0	0	1
CR8-RR	<addr>	1	1	0	1	1	1	1	0
Target									
TR1-NACK			1	0	0	0			
TR2-ACK			1	1	0	0			
TR3-RD	<data>		0		1	0	1		
TR4-RAD	<data>		1	1	1	0	1		
TR5-END			0		0	0			

Origins of Table

- Message exchange diagrams
  - Transitions were identified – CRx, TRx
- Description of transitions added
- Table of field values for transitions

Start Request

- R/W selection carried in DATA[0]
  - Should there be separate Start Read and Write?
- 10-bit addressing encoded in DATA[7:3]
  - Should there be a 10-bit address start variations?
- Should there be a separate Restart request?
  - Restart determined by current bus state

10-bit Address Request

- Does this need to be an explicit command?
  - The Write Continue request provides same info

End Request

- Separate Read and Write End Requests?
  - Read ends with NAK,Stop
  - Write ends with Stop
  - Target Controller will know which sequence to use

Read Response

- Currently have TR3-RD, TR4-RAD
  - Response Data
  - Response ACK and Data
- Controller can infer ACK from context

Recommended Request Encodings

- Start Request
  - R/W carried in Data[0]
  - Also used for Restart
- Write Request
  - Also used for 2nd byte of 10-bit addressing
- Read Acknowledge
- End Request
  - Will always generate an End Response

Recommended Response Encoding

- NACK Response
- ACK Response
  - Generated in response to Write Requests
  - Not generated after a read Start
- Read Response
- Stop Response