Initialisasi dan AntiCollision

Kasus pertama hanya menggunakan 1 PICC tanpa Collision dengan tipe double size, berikut prosesnya :

Keterangan: PCD = Reader, PICC = Card, UID = 04 DA E9 CA B5 28 80

1. PCD = REQA (mentransmisikan byte 0x26)
   PICC = ATQA (merespon PCD dengan 16bit yang isinya seperti gambar)

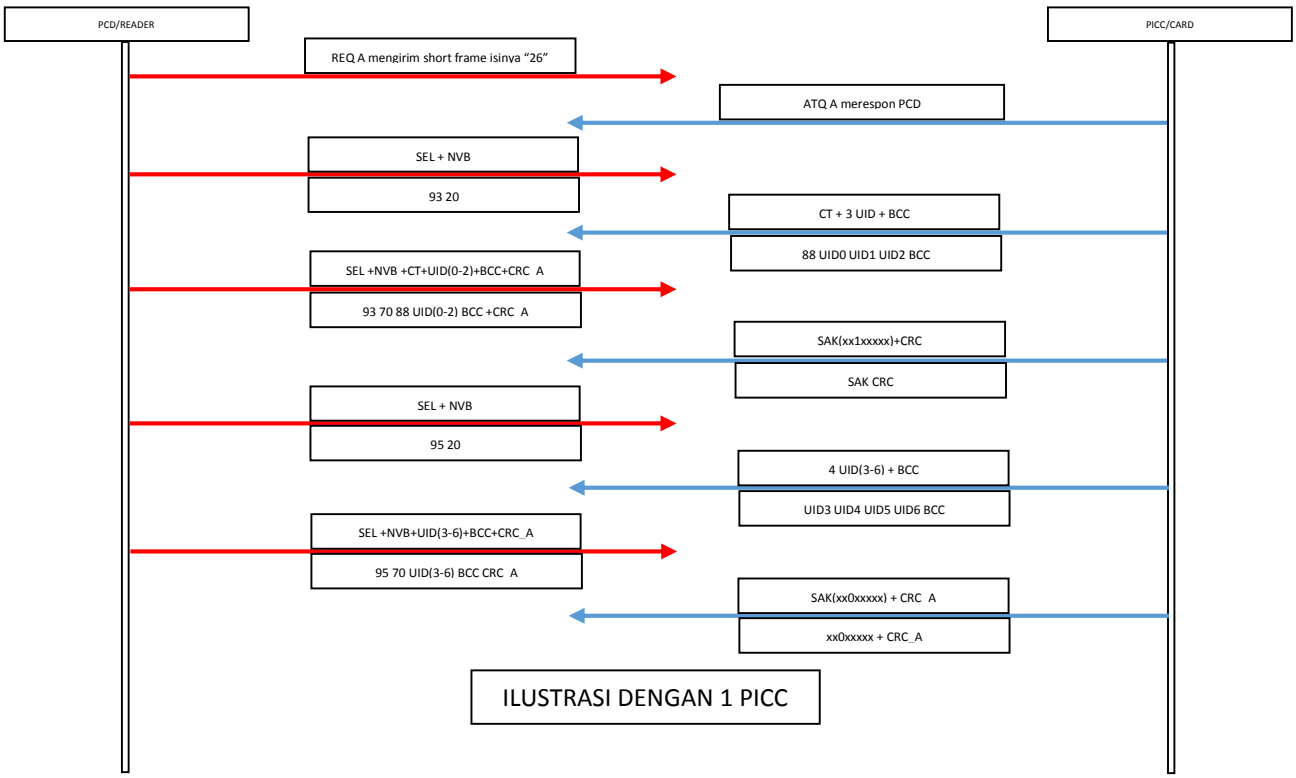
Table 4 — Coding of ATQA

MSB															LSB
b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1
RFU				Proprietary coding				UID size bit frame		RFU	Bit frame anticollision				

All RFU bits shall be set to 0.

Sehingga ATQA yang dikirimkan "0000 0000 0100 0100" (0x0044)

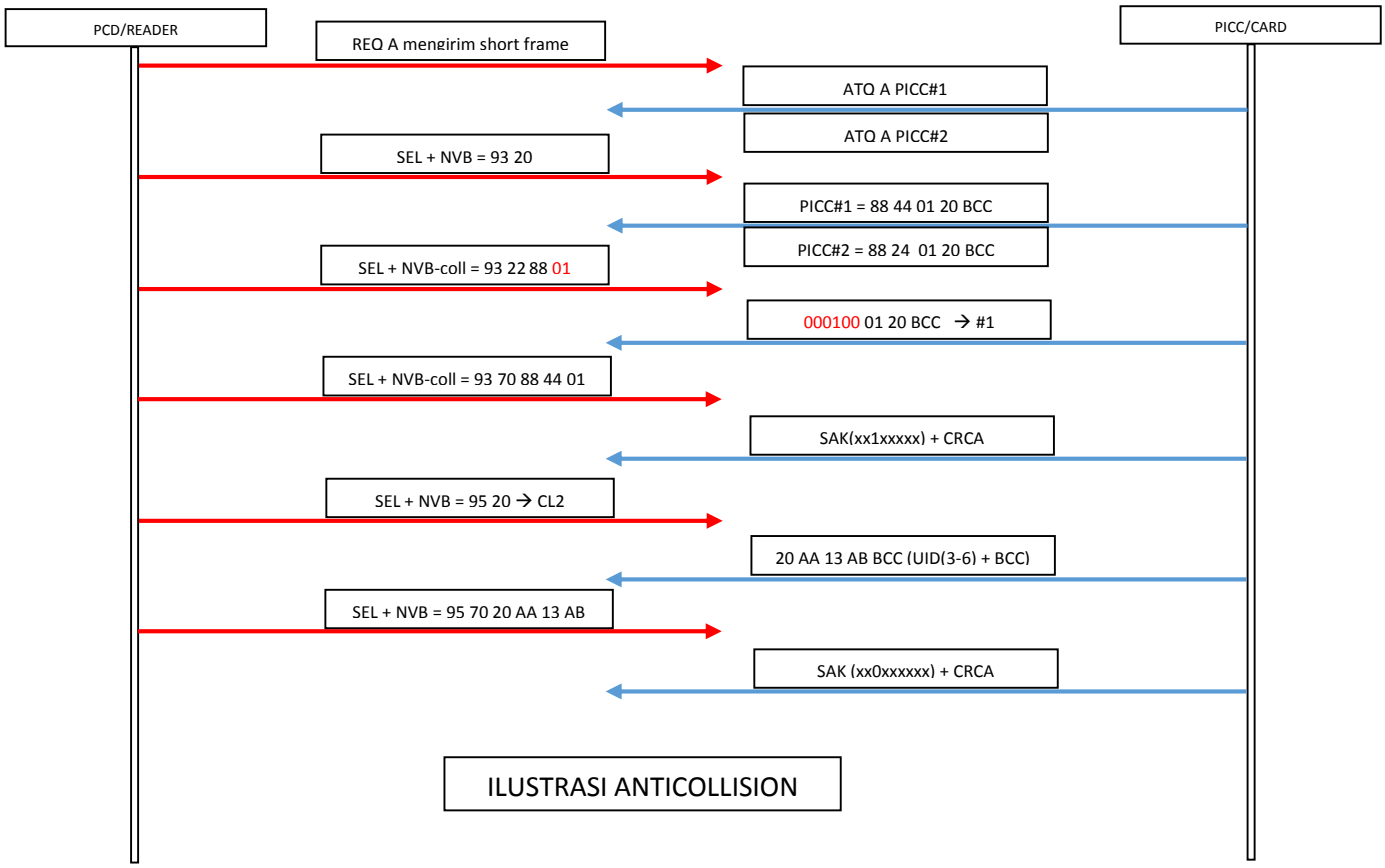
2. PCD = SEL + NVB (mentransmisikan commend Select Cascade Level 1(CL1) isinya "93 20")
   PICC = 88 04 DA E9 BF (merespon commend Select dengan : CT* + UID(0-2) + BCC**)
   *CT merupakan bit tambahan yang isinya 0x88
   **BCC merupakan hasil dari X-Or 4byte pertama (CT x-or $UID_{(0-2)}$)
3. PCD = SEL + NVB + CT + $UID_{(0-2)}$ + BCC + CRC_A (mentransmisikan commend Select yang isinya = 93 70 88 04 DA E9 BF CRC_A)
   PICC = SAK* + CRC_A (merespon dengan SAK CRC_A yang isinya : xx1xxxxx + CRC_A)
   * SAK = xx0xxxxx pengiriman UID selesai atau xx1xxxxx pengiriman UID ada lanjutan
   Note: Karena PICC ini merupakan tipe double size maka akan dilakukan pengiriman 4byte UID berikutnya, perhatikan SAK yang dikirim PICC menunjukkan bahwa ada lanjutan UID lagi.
4. PCD = SEL + NVB (mentransmisikan commend Select Cascade Level 2(CL2) isinya = "95 20")
   PICC = CA B5 28 80 D7 (merespon commend Select Dengan $UID_{(3-6)}$ + BCC)
5. PCD = SEL + NVB + $UID_{(3-6)}$ + BCC + CRC_A (mentransmisikan commend Select yang isinya = 95 70 CA B5 28 80 D7 CRC_A)
   PICC = SAK + CRC_A (merespon dengan SAK CRC_A yang isinya : xx0xxxxx + CRC_A)
   Note: Pengiriman UID telah selesai dan akan dilanjutkan ke proses selanjutnya yaitu, ISO_IEC14443-4



Kemudian kasus kedua yaitu dengan Collision menggunakan 2 PICC tipe double size, berikut prosesnya:

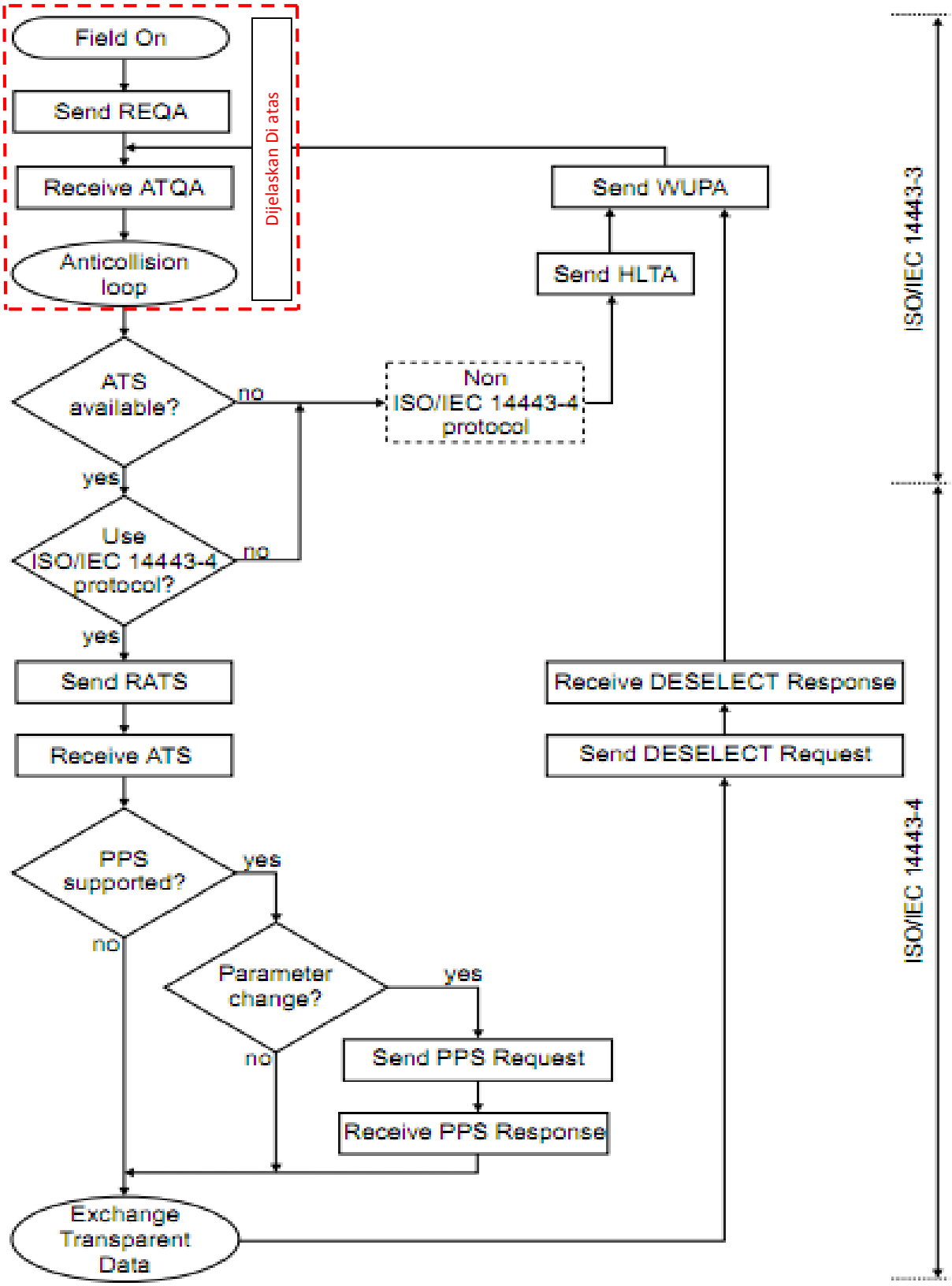
Keterangan : UID#1 = 14 DA E9 CA B5 28 80 UID#2 = 03 DA E9 CA B5 28 80

1. PCD = REQA (mentransmisikan byte 0x26)
   PICC = ATQA#1 (merespon PCD dengan 16bit = "0000 0000 0100 0100" (0x0044))
   ATQA#2 (merespon PCD dengan 16bit = "0000 0000 0100 0010" (0x0042))
2. PCD = SEL + NVB (93 20)
   PICC = #1 ➔CT + $UID_{(0-2)}$ + BCC (88 14 DA E9 AF)
   #2 ➔CT + $UID_{(0-2)}$ + BCC (88 03 DA E9 B8)
   Terjadi collision pada bit ke-4 UID0 sehingga PCD menghandle dengan anticollision
3. PCD = SEL + NVB-coll + CT + 0001 (mentransmisikan Select commend yang isinya = 93 24 88 0001 "dikirin NVB 24 karena terjadi coll di bit ke-4 UID0")
   PICC = #1 ➔0100 DA E9 AF
4. PCD = SEL + NVB + CT + $UID_{(0-2)}$ + BCC +CRC_A(mentransmisikan commend Select yang isinya 93 70 88 14 DA E9 AF CRC_A)
   PICC = SAK* + CRC_A (merespon dengan SAK CRC_A yang isinya : xx1xxxxx + CRC_A)
   * SAK = xx0xxxxx pengiriman UID selesai atau xx1xxxxx pengiriman UID ada lanjutan
   Note: akan melanjutkan ke bagian UID berikutnya
5. PCD = SEL + NVB (mentransmisikan commend Select Cascade Level 2(CL2) isinya = "95 20")
   PICC = CA B5 28 80 D7 (merespon commend Select Dengan $UID_{(3-6)}$ + BCC)
6. PCD = SEL + NVB + UID(3-6) + BCC + CRC_A (mentransmisikan commend Select yang isinya = 95 70 CA B5 28 80 D7 CRC_A)
   PICC = SAK + CRC_A (merespon dengan SAK CRC_A yang isinya : xx0xxxxx + CRC_A)



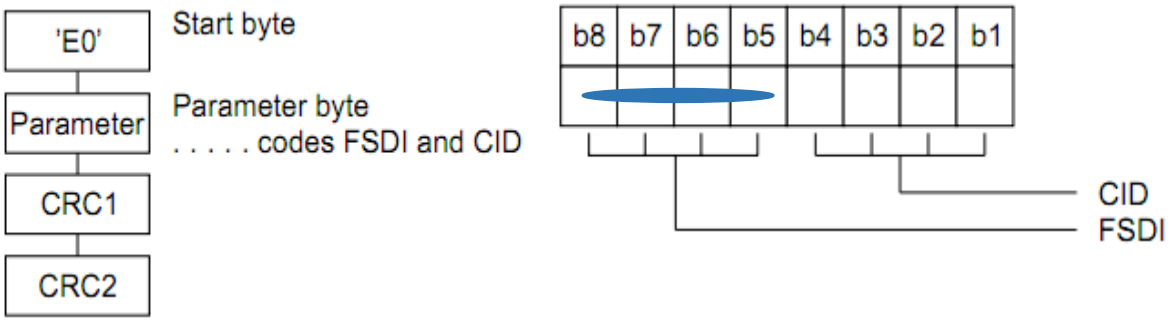
ILUSTRASI ANTICOLLISION

Transmission Protocol



Setelah proses anticollision selesai maka, PCD akan memberikan perintah selanjutnya berupa command yaitu:

1. PCD = RATS (mentransmisikan command RATS dengan format byte sebagai berikut:)

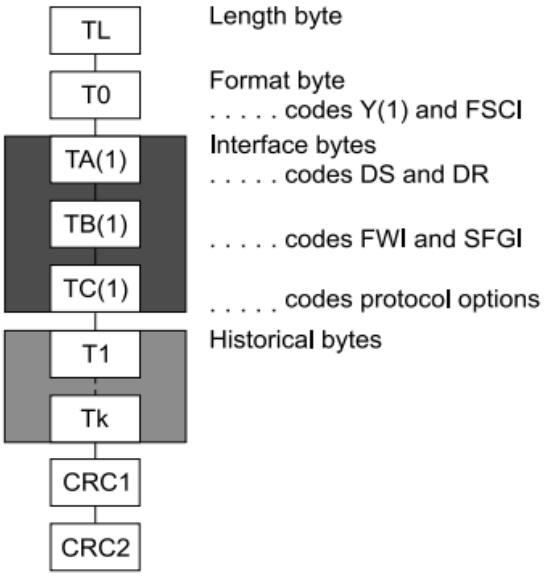


FSDI	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'-'F'
FSD (bytes)	16	24	32	40	48	64	96	128	256	RFU >256

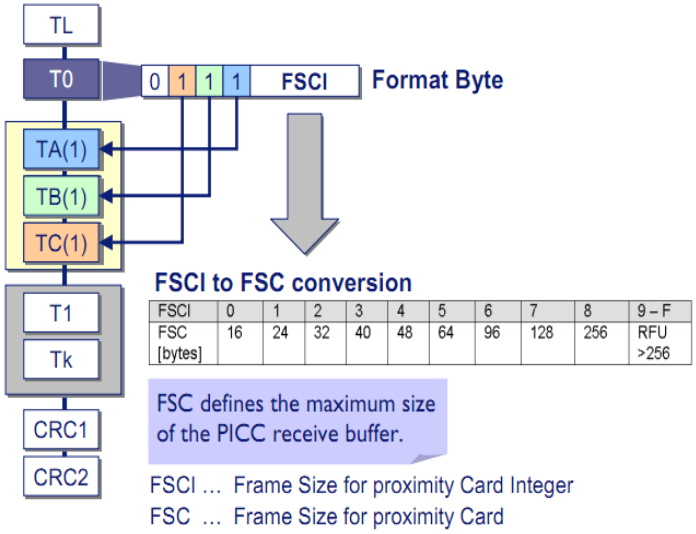
Untuk byte pertama dari command RATS merupakan ***start byte*** yang isinya 0xE0 kemudian di ikuti oleh parameter yang 4 bit MSB berisi FSDI. FSDI merupakan parameter untuk mendefinisikan ukuran frame maksimum yang dapat di terima oleh PCD. Panjang data yang dapat diterima ditentukan oleh codingan FSD (Contoh: FSDI = 0111 maka panjang frame dalam satu penerimaan = 128byte).

Sedangkan 4 bit LSB merupakan CID. CID adalah jumlah logika yang dialamatkan PICC dalam kisaran 0-14 sedangkan 15 merupakan RFU. Nilai dari CID harus unik untuk setiap PICC dalam keadaan aktif pada waktu yang bersamaan. (belum paham)

PICC = ATS (merespon command dari PCD yang merupakan RATS dengan format ATS sebagai berikut:)



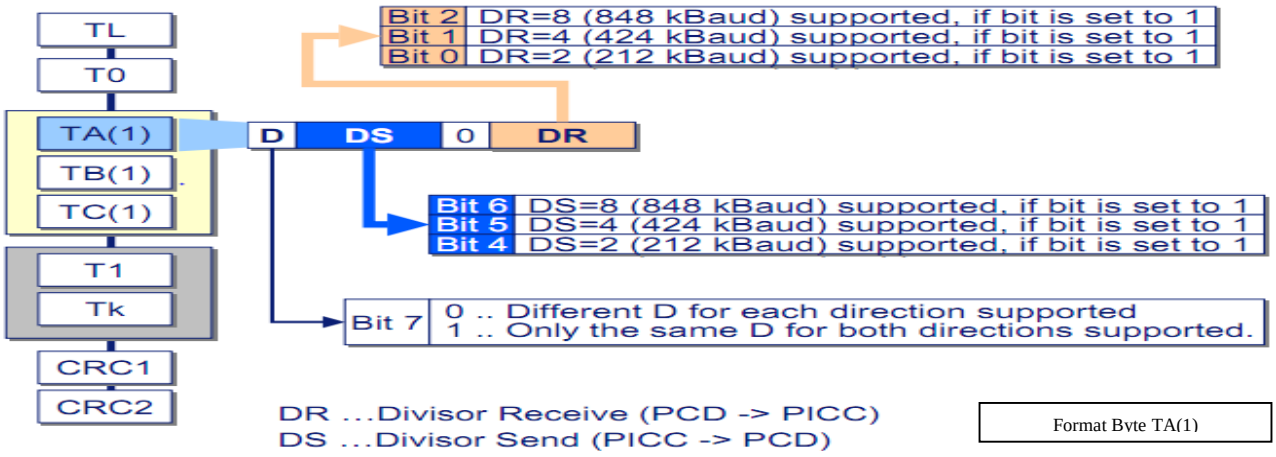
gambar di samping merupakan format byte ATS yang di transmisikan PICC ke PCD.



TL mendefinisikan panjang byte yang di transmisikan sehingga dari format di atas TL = 0x06 yangartinya ada 6 byte data yang akan di trasnmisikan tanpa menghitung byte CRC (gambar kiri). Sedangkan isi dari format byte T0 yaitu, seperti gambar (kanan) 4 bit MSB menggambarkan b8 selalu di set 0 karena jika 1 merupakan

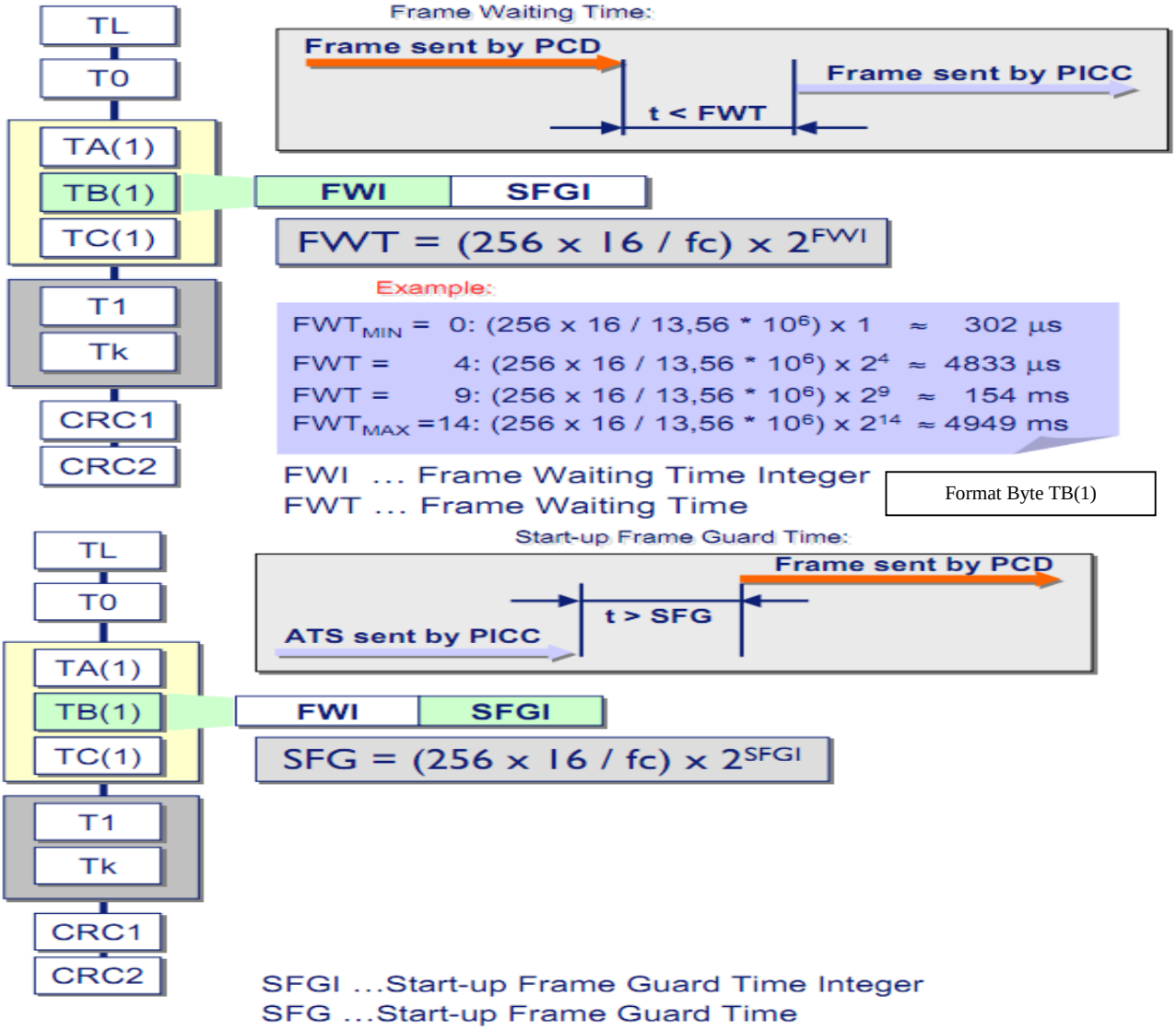
RFU. Sedangkan b7-b5 menandakan TA(1), TB(1) dan TC(1) telah di transmit jika di set 1.Untuk format byte TA(1), TB(1) dan TC(1) sebagai berikut:

Interface Byte TB(1)



Format Byte TA(1)

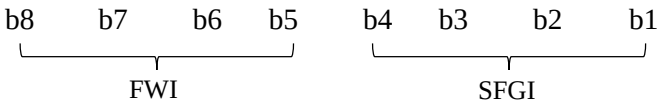
Untuk penjelasan dari masing-masing bit TA(1), yaitu: D di set 0 mendukung baud rate yang berbeda antara PCD dan PICC, diset 1 hanya mendukung baud rate yang sama antara PCD dan PICC. DS(b6-b4) di set sesuai baud rate PICC ke PCD (lihat gambar). Sedangkan untuk b3 di set 0 jika di set 1 merupakan RFU, sedangkan DR (b2-b0) merupakan baud rate PCD ke PICC. Untuk DR dan DS defaultnya 000.



Format Byte TB(1)

Interface Byte TB(1)

TB(1) menyampaikan informasi untuk menentukan waktu tunggu frame (FWT) dan waktu pengaktifan frame guard time (SFGT).



SFGT adalah waktu yang dibutuhkan oleh PICC sebelum PICC siap untuk menerima frame berikutnya setelah PICC mengirim ATS. SFGT didapat dari SGFI. Dimana default SFGI = 0.

SFGT = (256 x 16 /fc) x 2^{SFGI} ..... range SFGI 0 – 14

$SFGT_{max}$ = ~ 4949 ms

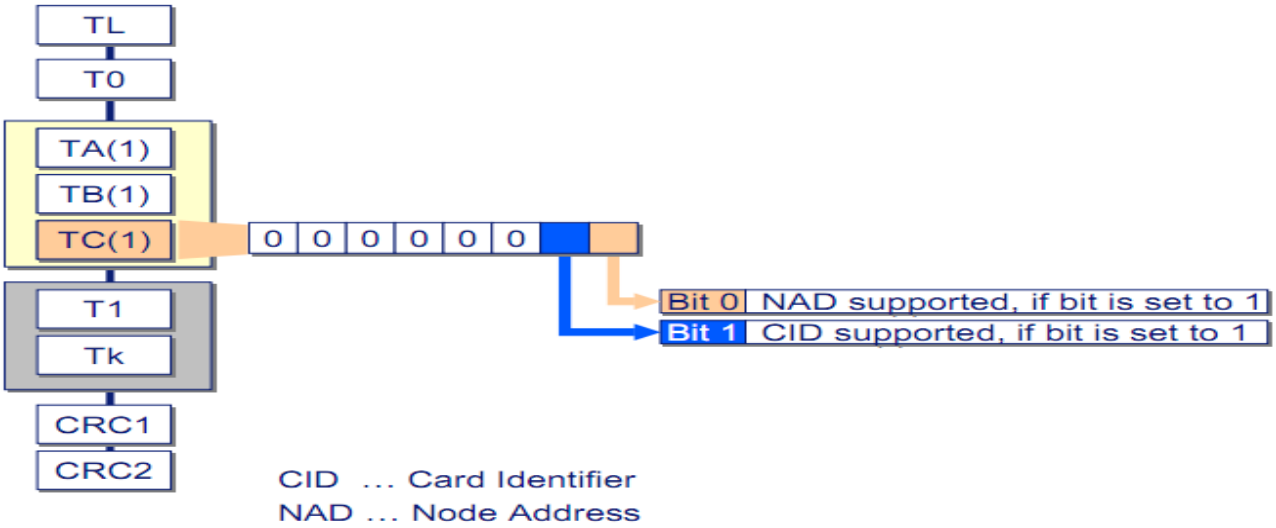
FWI (Frame Writing time Integer) digunakan untuk mendefinisikan FWT. FWT adalah waktu maksimal untuk PICC memulai responnya setelah berakhirnya frame PCD.

FWT = (256 x 16 /fc) x 2^{FWI} .... range FWI 0 – 14

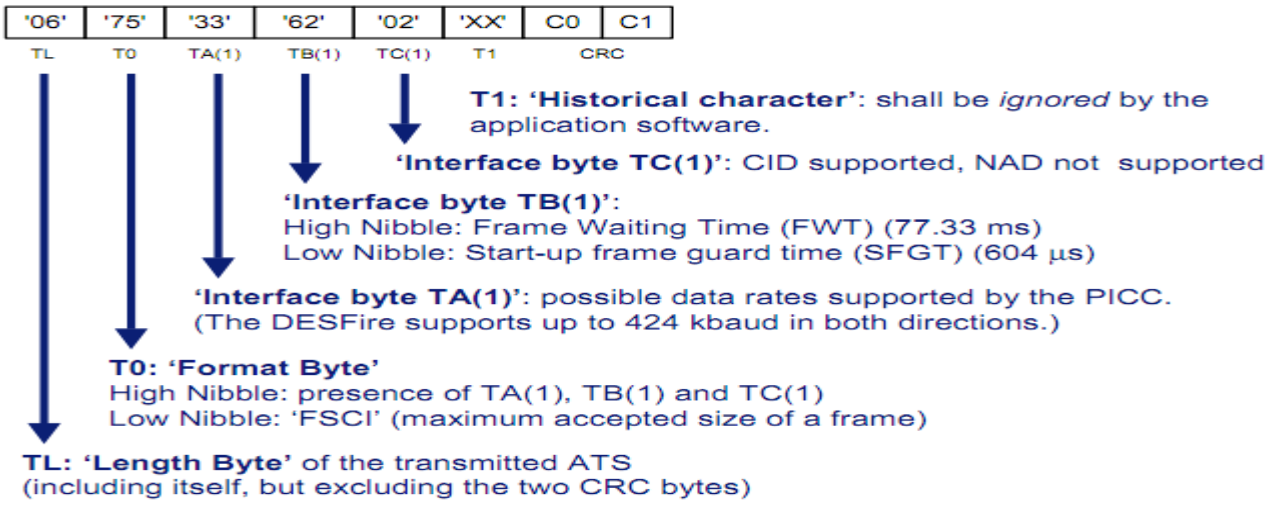
Untuk FWI = 0, FWT minimal (~302 us)

FWI = 14, FWT maksimal (~4949 ms)

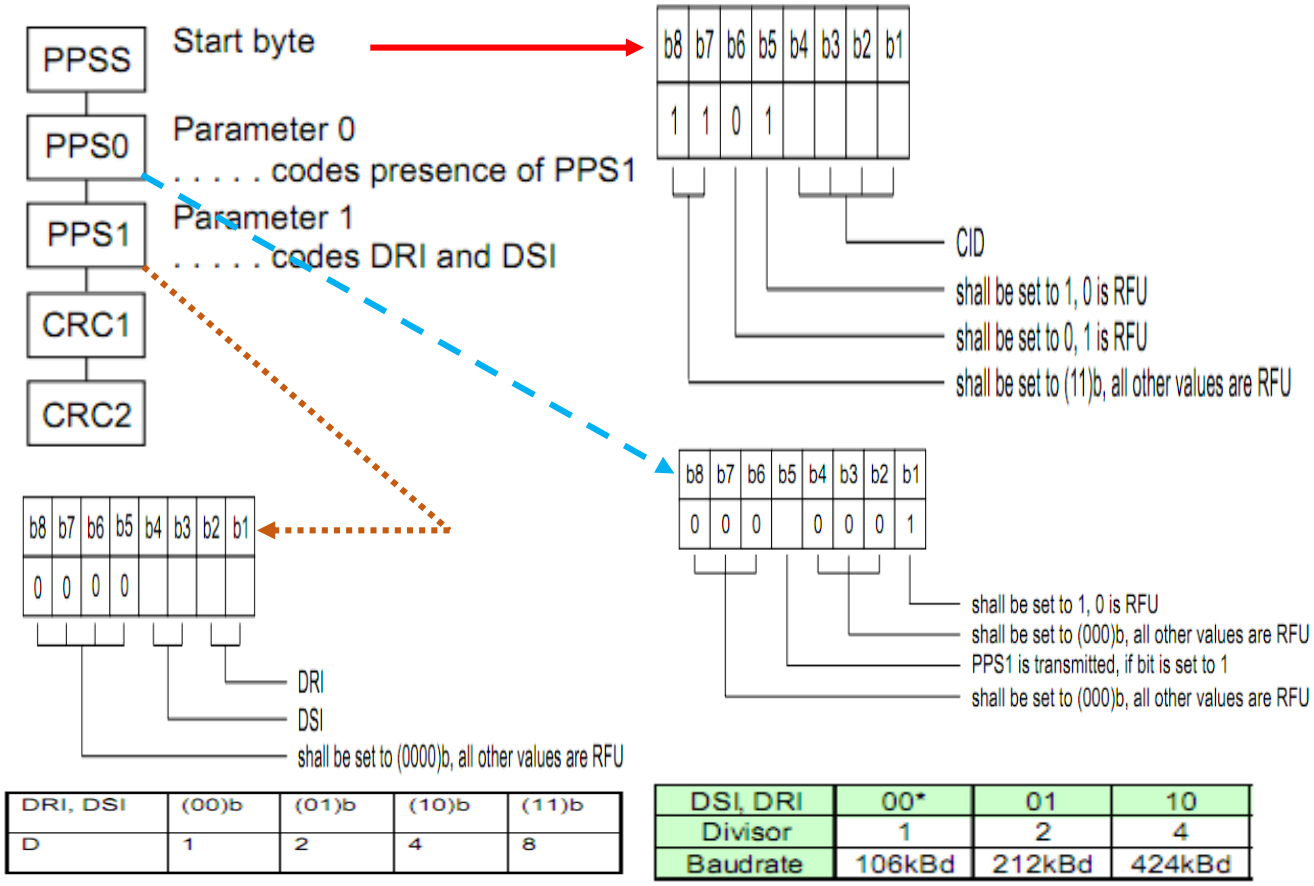
Interface Byte TC(1)



Secara keseluruhan contoh dari byte ATS adalah sebagai berikut:



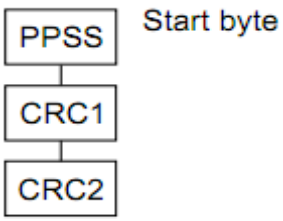
2. PCD = PPS request (mentransmisikan command PPS yang isinya sebagai berikut:)



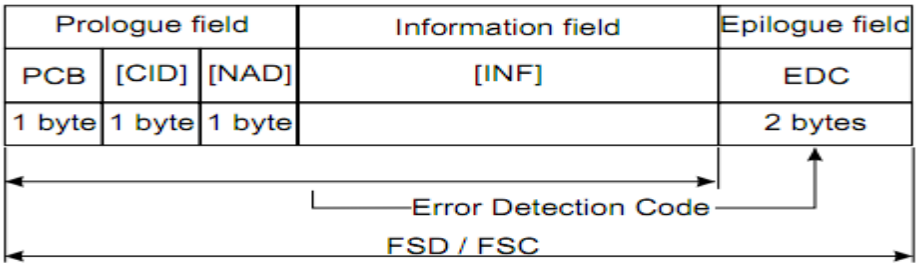
DRI, DSI	(00)b	(01)b	(10)b	(11)b
D	1	2	4	8

DSI, DRI	00*	01	10
Divisor	1	2	4
Baudrate	106kBd	212kBd	424kBd

PICC = PPS respon (merespon command dari PPS req dari PCD yang isinya sebagai berikut:)



Setelah proses di atas di lakukan, maka memasuki proses Block Format dengan format block sebagai berikut:



Untuk coding byte PCB ada 3 jenis, yaitu:

a) I-block bertujuan untuk menyampaikan informasi yang digunakan oleh application layer
b) R-block bertujuan untuk menyampaikan ACK atau NACK. R-block tidak pernah berisi INF.
c) S-block digunakan untuk kontrol pertukaran informasi antara PICC dan PCD. Ada 2 tipe S-block
    1. WTX (waiting time extension) berisi 1 byte informasi.
    2. DESELECT tidak berisi informasi.

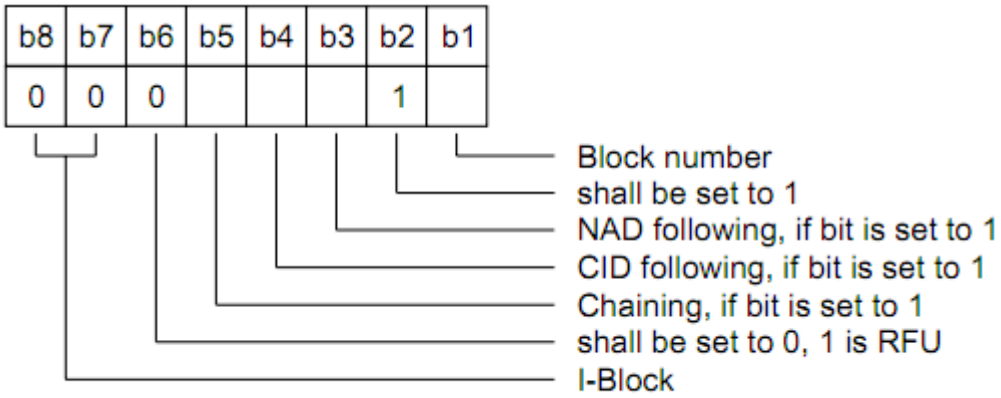


Figure 15 — Coding of I-block PCB

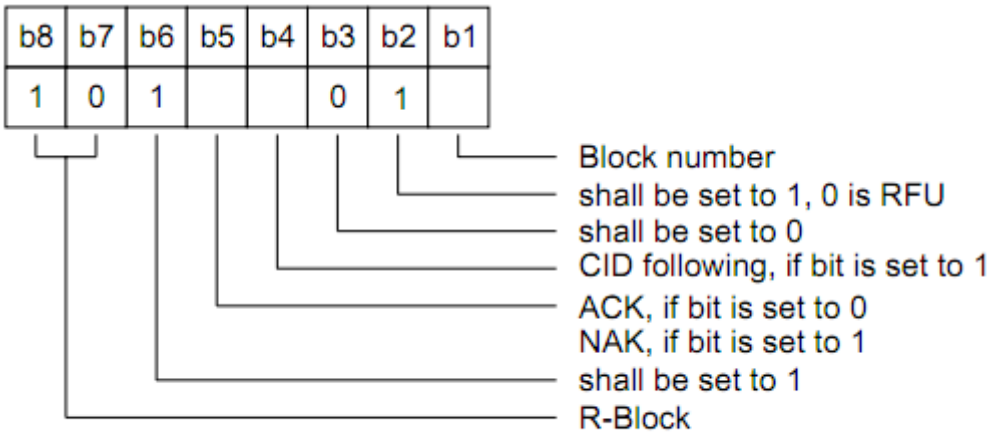


Figure 16 — Coding of R-block PCB

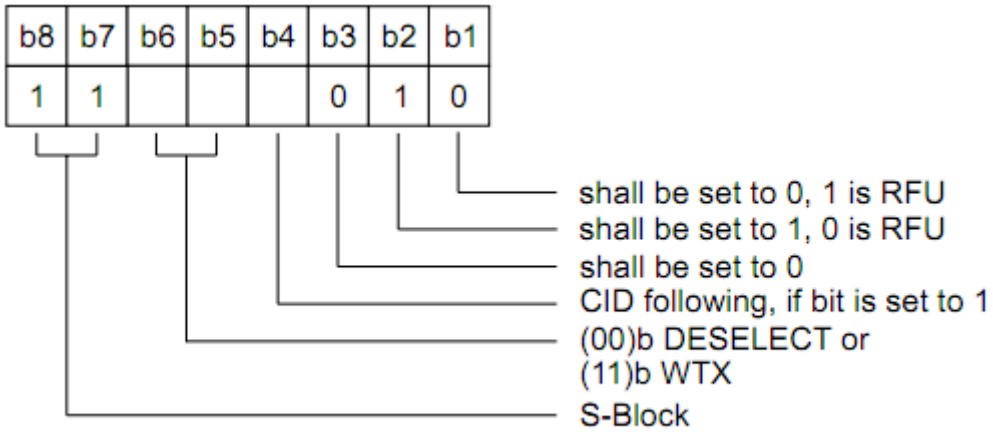


Figure 17 — Coding of S-block PCB

Tiga gambar di atas menggambarkan format coding dari I, R dan S block. Yang menentukan jenis blocknya adalah b8 dan b7. Dan masing-masing block memiliki fungsi yang berbeda-beda.

Note: FWT mendefinisikan waktu jeda saat PICC akan memulai respon framenya setelah pengiriman terakhir frame PCD ke PICC(perhitungan dijelaskan di TB(1)). Sedangkan jika PICC tidak memberikan respond dalam rentang waktu FWT PCD berhak mengirim kembali informasinya(framenya). Namun dalam satu kasus PICC merespon dengan FWTX yaitu, frame INF yang berisi permintaan waktu tambahan.

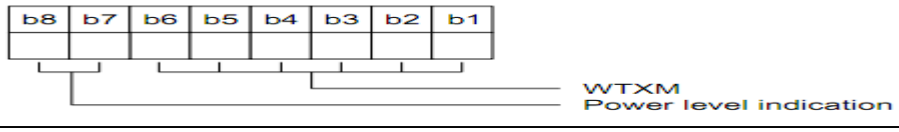
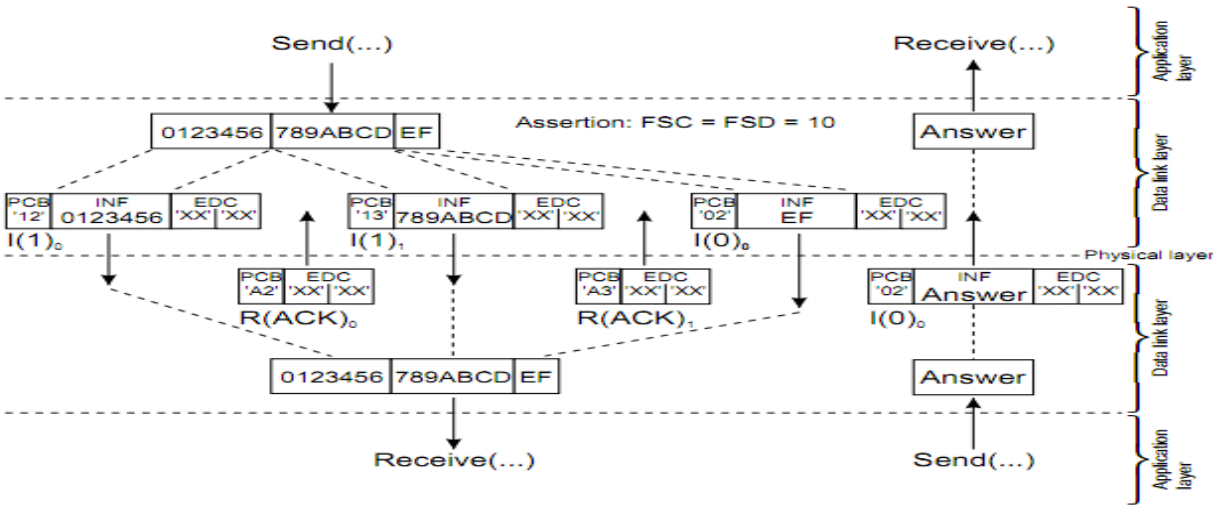


Table 3 — Coding of power level indication

(00)b	PICC does not support the power level indication
(01)b	Insufficient power for full functionality
(10)b	Sufficient power for full functionality
(11)b	More than sufficient power for full functionality

Untuk rule berikutnya harus dibaca lebih detail maksud dan tujuan dari rule PICC dan PCD ada pada ISO/IEC 14443-4.



NOTE This example does not use the optional fields NAD and CID.

Figure 22 — Chaining