عنوان پروژه:

# RFID Anticollision

دانشجو:

**سعید احمدی**

تابستان **۱۳۹۱**

## پیشگفتار

من در این پروژه به طراحی وساخت یک سیستم RFID از نوع مایفر که موجود در بازار می باشد , پرداختم . برای بخش میکروپروسسوری این پروژه از میکرو کنترلر استفاده شده است و همچنین از کارت های مایفر A , S50 که دارای 1k حافظه داخلی می باشند به عنوان تگ مورد استفاده قرار گرفته اند .

من این پروژه را در چهار فصل تهیه کرده ام که در فصل اول به توضیح مسائل کلی و انواع و اقسام این گونه سیستم ها و همچنین کاربرد های این سیستم پرداخته , در فصل دوم به سخت افزارهای استفاده شده در این پروژه و مشخصات فنی هر سخت افزار و چگونگی کارکرد هر کدام و تشریح نقشه فنی سیستم می پردازم . در فصل سوم نیز به توضیح برنامه نوشته شده در قسمت میکروکنترلر سیستم که مرکز فرماندهی سیستم را تشکیل می دهد پرداخته ام . در فصل آخر به طور مخصر پروژه را مورد بررسی قرار داده و نتایج و برخی کاربرد های آن را ارائه خواهم کرد .

امروزه ضرورت شناسایی خودکار عناصر و جمع آوری داده مرتبط به آنان بدون نیاز به دخالت انسان جهت ورود اطلاعات در بسیاری از عرصه های صنعتی ، علمی ، خدماتی و اجتماعی احساس می شود .در پاسخ به این نیاز تاکنون فناوری های متعددی طراحی و پیاده سازی شده است .

به مجموعه ای از فناوری ها که از آنان برای شناسایی اشیاء ، انسان و حیوانات توسط ماشین استفاده می گردد ، شناسایی خودکار و یا به اختصار Auto ID گفته می شود . هدف اکثر سیستم های شناسایی خودکار ، افزایش کارآیی ، کاهش خطاء ورود اطلاعات و آزاد سازی زمان کارکنان برای انجام کارهای مهمتر نظیر سرویس دهی بهتر به مشتریان است .

تاکنون فناوری های مختلفی به منظور شناسایی خودکار طراحی و پیاده سازی شده است . کدهای میله ای ، کارت های هوشمند ، تشخیص صدا ، برخی فناوری های بیومتریک ، OCR ( optical character recognition ) و RFID (identification radio frequency ) نمونه هایی

در این زمینه می باشند . در اینجاست که RFID نقش خود را به عنوان یک تکنولوژی جدید ولی ضروری برای جوامع امروز پیدا می کند . از این رو بهبود و پیشرفت این تکنولوژی یک امر ضروری به نظر می رسد .

RFID با استفاده از ارتباطات مبتنی بر فرکانس های رادیویی امکان شناسایی خودکار ، ردیابی و مدیریت اشیاء ، انسان و حیوانات را فراهم می نماید . عملکرد RFID وابسته به دو دستگاه تگ و کدخوان است که جهت برقراری ارتباط بین یکدیگر از امواج رادیویی استفاده می نمایند.

به مجموعه ای از فناوری ها که در آنان برای شناسایی خودکار افراد و اشیاء از امواج رادیویی استفاده می گردد ، RFID گفته می شود . از روش های مختلفی برای شناسایی افراد و اشیاء استفاده می شود. ذخیره شماره سریال متناسب به یک فرد و یا شی درون یک ریزتراشه که به آن یک آنتن متصل شده است ، یکی از متداولترین روش های شناسایی خودکار است .
به تلفیق تراشه و آنتن ، تگ RFID و یا فرستنده خودکار RFID گفته می شود . تراشه به کمک آنتن تعبیه شده ، اطلاعات لازم جهت شناسایی آیتم مورد نظر را برای یک کدخوان ارسال می نماید . کدخوان امواج رادیویی برگردانده شده از تگ RFID را به اطلاعات دیجیتال تبدیل می نماید تا در ادامه ، امکان ارسال داده برای کامپیوتر و پردازش آن فراهم گردد.

RFID یک تکنولوژی مهم جهت شناسایی اشیاء ، جمع آوری داده و مدیریت اشیاء را ارائه می نماید . تکنولوژی فوق مشتمل بر مجموعه ای از فناوری های حامل داده و محصولاتی است که به مبادله داده بین حامل و یک سیستم مدیریت اطلاعات از طریق یک لینک فرکانس رادیویی کمک می نماید . تگ های RFID با استفاده از یک فرکانس و بر اساس نیاز سیستم ( محدوده خواندن و محیط ) ، پیاده سازی می گردند . تگ ها به صورت فعال ( به همراه یک باطری ) و یا غیرفعال ( بدون باطری ) پیاده سازی می شوند . تگ های غیرفعال، توان لازم جهت انجام عملیات را از میدان تولید شده توسط کدخوان می گیرند .

کدخوان RFID ، معمولا به یک کامپیوتر متصل می شود و دارای نقشی مشابه با یک اسکنر کد میله ای است . مسئولیت برقراری ارتباط لازم بین سیستم اطلاعاتی و تگ های RFID برعهده کدخوان RFID است.

از محدودیت های این سیستم می توان به این موضوع اشاره کرد که به دلیل تازگی این تکنولوژی یک مقدار گران تلقی می شود و همچنین اینکه در کشور فقط تعداد و انواع محدودی از قطعات این تکنولوژی وجود دارد و به گونه ای دست طراح را در طراحی هر گونه مداری باز نگذاشته است . غیر از این مشکل , محدودیت و معایب دیگری نیز وجود دارد که در زیر به برخی از آنها می پردازم .

**تداخل:** به دو صورت اتفاق می‌افتد:

تداخل Reader ها: زمانی اتفاق می‌افتد که سیگنال‌های ارسال شده از چند دستگاه Reader تداخل پیدا می‌کنند.

تداخل tag ها: زمانی اتفاق می‌افتد که تعداد tag های بسیار زیادی در فضای کوچکی وجود داشته باشند.

**مسئله ایمنی:** اکثر tag های RFID حتی پس از خرید و خروج از فروشگاه فعال هستند. در نتیجه اطلاعات آن‌ها می‌تواند توسط دستگاه‌های Reader خوانده شود. بنابراین احتمال سرقت کالاها افزایش می‌یابد. علاوه بر این بسیاری از سازمان‌ها به هنگام خرید مشتری اطلاعاتی را درباره مشتری(از جمله شماره Credit-card، آدرس،نام و... به RFID tag کالاها منتقل می‌کنند تا لیستی از مشخصات مشتریان خود داشته باشند؛ که این امر اطلاعات محرمانه مشتریان را به خطر می‌اندازد. موضوع امنیت زمانی بیشتر به چشم می‌خورد که به کاربردهای RFID در پزشکی توجه کنیم.

**مشکلات اجتماعی:** بر اثر پیشرفت‌های اخیر تکنولوژی در بسیاری از مناطق دنیا از جمله فرانسه، نروژ و... فن‌آوری RFID برای تشخیص هویت افراد استفاده می‌شود. این امر به عقیده بسیاری از افراد

نامطلوب است؛ زیرا شخصیت اجتماعی و انسانی آن‌ها را زیرسوال برده و سبب می‌شود که به انسان‌ها به چشم یک ربات نگریسته شود.

**عدم وجود استانداردها:** شرکت‌های متعددی وجود دارند که دستگاه‌های RFID را تولید می‌کنند.اما قوانین و استانداردهای جهانی خاصی برای این تولید وجود ندارد.این مسئله سبب می‌شود که فن‌آوری RFID طراحی شده برای یک کمپانی یا شرکت تنها در همان شرکت قابل استفاده باشد و tag های موجود در روی محصولات یک کمپانی(مثلا تامین‌کننده) ممکن است توسط کمپانی دیگر خوانده نشود که این امر مشکلات فراوانی را ایجاد می کند.استانداردهای موجود برای تعیین فرکانس RFID نیز در کشورهای مختلف تفاوت دارد.

# فهرست مطالب

**پیش گفتار** ..... **i**

**فصل اول عنوان پروژه و تحقیق** ..... **۱**

۱.۱ عنوان پروژه ..... ۱
۱.۲ تعریف پروژه ..... ۱
۱.۳ تاریخچه RFID ..... ۲
۱.۴ اصول فناوری RFID ..... ۲

**فصل دوم سیستم های RFID ضدتداخل** ..... **۱۵**

۲.۱ مقدمه ..... ۱۵
۲.۲ معرفی سیستم های RFID ضد تداخل ..... ۱۵
۲.۳ پروتکل های ضد تداخل برای سیستم های RFID غیرفعال ..... ۱۷
۲.۳.۱ زمان بندی دسترسی چندگانه (TDMA) ..... ۱۷
۲.۳.۲ کدبندی دسترسی چندگانه (CDMA) ..... ۱۹
۲.۳.۳ مکان یابی دسترسی چندگانه (SDMA) ..... ۱۹
۲.۳.۴ فرکانس بندی دسترسی چندگانه (FDMA) ..... ۲۰
۲.۴ بررسی کامل پروتکل TDMA ..... ۲۱
۲.۴.۱ الگوریتم های ضد تداخل بر اساس ALOHA ..... ۲۲
۲.۴.۲ الگوریتم ALOHA پایه و Slotted ALOHA ..... ۲۳
۲.۴.۳ الگوریتم BFSA (Basic Frame Slotted ALOHA) ..... ۲۴
۲.۴.۴ الگوریتم DFSA (Dynamic Frame Slotted ALOHA) ..... ۲۵
۲.۴.۵ الگوریتم ADFSA(Advanced Dynamic Frame Slotted ALOHA) ..... ۲۶
۲.۴.۶ گروه بندی تعداد تگ ها ی پاسخ دهنده ..... ۲۸
۲.۴.۷ الگوریتم Tree Slotted ALOHA ..... ۲۹
۲.۴.۸ الگوریتم های مبتنی بر درختی ..... ۳۰
۲.۴.۹ الگوریتم های درختی باینری ..... ۳۰
۲.۴.۱۰ الگوریتم های Query Tree ..... ۳۱

۲.۴.۱۱ س الگوریتم های مبتنی بر Tree – ALOHA ..... ۳۲
۲.۴.۱۲ الگوریتم Slotted Binary Tree ..... ۳۳
۲.۴.۱۳ الگوریتم ضد تداخل مبتنی بر Bi-slotted tree ..... ۳۳
۲.۴.۱۴ الگوریتم درخت پرس و جوی قاب بندی شده (FQT) framed query true ..... ۳۶
۲.۴.۱۵ الگوریتم (QT-ALOHA) ..... ۳۷
۲.۴.۱۶ الگوریتم FS-ALOHA با برآورد تگ و جداسازی (EBFSA) ..... ۳۹
۲.۴.۱۷ الگوریتم FSA با فریم راهنما و انتخاب باینری (FSAPB) ..... ۴۱

**فصل سوم بررسی استاندارد ISO/IEC 14443 ..... ۴۶**

مقدمه ..... ۴۶

**استاندارد 14443-1 ..... ۴۷**

۱. هدف ..... ۴۷
۲. قوانین ..... ۴۷
۳. تعاریف ، اختصارها و نمادها ..... ۴۸
۳.۱ تعاریف ..... ۴۸
۴. ویژگی های فیزیکی ..... ۴۹
۴.۱ کلیت ..... ۴۹
۴.۲ ابعاد ..... ۴۹
۴.۳ ویژگی های اضافه ..... ۵۰

**استاندارد 14443-2 ..... ۵۱**

۱. هدف ..... ۵۱
۲. قوانین ..... ۵۱
۳. تعاریف و قوانین ..... ۵۲
۴. اختصارات و نمادها ..... ۵۲
۵. رابطه اولیه برای کارت ها ..... ۵۳
۶. انتقال قدرت ..... ۵۳
۷. سیگنال رابط ..... ۵۴
۸. سیگنال ارتباطی نوع A ..... ۵۵
۸.۱ ارتباطات PCD به PICC ..... ۵۵
۸.۲ ارتباطات PICC به PCD ..... ۵۸
۹. سیگنال ارتباطی نوع B ..... ۶۰

۹.۱ ارتباط PCD با PICC ..... ۶۰
۹.۱.۱ نرخ داده ..... ۶۰
۹.۱.۲ مدولاسیون ..... ۶۰
۹.۱.۳ کدگذاری بیت ..... ۶۱
۹.۲ ارتباط PICC با PCD ..... ۶۱
۹.۲.۱ نرخ داده ..... ۶۱
۹.۲.۲ بار مدولاسیون ..... ۶۲
۹.۲.۳ ساب کریر ..... ۶۲
۹.۲.۴ مدولاسیون ساب کریر ..... ۶۲
۹.۲.۵ کدگذاری و نمایش بیت ..... ۶۳
۱۰. حداقل محدوده کوپلینگ PICC ..... ۶۴
**استاندارد 14443-3 ..... ۶۵**
۱. هدف ..... ۶۵
۲. قوانین ..... ۶۵
۳. عبارات و معانی ..... ۶۶
۴. نماد ها ..... ۶۷
۵. رای گیری ..... ۶۹
۶. نوع A – نصب و ضدتداخل ..... ۶۹
۶.۱ بایت ، فریم و قالب دستور و زمانبندی ..... ۷۰
۶.۲ حالت های PICC ..... ۷۶
۶.۳ دستورات ..... ۷۸
۶.۴ مراحل انتخاب ..... ۸۰
۶.۴.۱ فلوچارت مراحل انتخاب ..... ۸۱
۶.۴.۲ پاسخ به درخواست ATQA ..... ۸۲
۶.۴.۳ ضدتداخل و انتخاب ..... ۸۳
**استاندارد 14443-4 ..... ۹۰**
۱. هدف ..... ۹۰
۲. قوانین ..... ۹۰
۳. تعاریف و عبارات ..... ۹۱
۴. نماد ها ..... ۹۲
۵. فعال کردن پروتکل PICC نوع A ..... ۹۳
۵.۱ درخواست برای پاسخ به انتخاب (RATS) ..... ۹۴
۵.۲ پاسخ به انتخاب ATS ..... ۹۵
۵.۳ زمان انتظار فریم فعال سازی ..... ۹۸
۵.۴ شناسایی و بازیابی خطا ..... ۹۹

۶. غیر فعال کردن پروتکل PICC نوع A ..... ۱۰۰
**فصل چهارم نمونه عملی RFID ضدتداخل ..... ۱۰۲**

۴.۱ مقدمه ..... ۱۰۲
۴.۲ نحوه کار مدار ..... ۱۰۳
۴.۳ بررسی اجزا مدار پردازشگر ..... ۱۰۳
۴.۴ بررسی نمای شماتیک و PCB مدار پردازشگر ..... ۱۰۴
۴.۵ ماژول YLMF018 ..... ۱۰۵
۴.۵.۱ استاندارد Mifare ..... ۱۰۶
۴.۵.۲ ویژگی های فنی ..... ۱۰۶
۴.۵.۳ تنظیمات ارتباط ..... ۱۰۷
۴.۵.۴ تصویر و ابعاد ماژول ..... ۱۰۷
۴.۵.۵ ویژگی ماژول ..... ۱۰۹
۴.۵.۶ پروتکل ارتباطی ..... ۱۰۹
۴.۵.۷ دستورات ..... ۱۱۰
۴.۶ برنامه میکرو ..... ۱۱۲
**فصل پنجم خلاصه پروژه و نتایج بدست آمده ..... ۱۴۰**

# فهرست اشکال و نمودارها

**شکل ۱-۱)** نمونه یک Tranceiver ..... ۲

**شکل ۱-۲)** طیف فرکانسی ..... ۱۲

**شکل ۲-۱)** الگوریتم های احتمالی ..... ۲۳

**شکل ۲-۲)** نمونه الگوریتم BFSA ..... ۲۴

**شکل ۲-۳)** مقایسه الگوریتم ها ..... ۲۸

**شکل ۲-۴)** الگوریتم های قطعی ..... ۲۹

**شکل ۲-۵)** نمونه الگوریتم FQT ..... ۳۷

**شکل ۲-۶)** نمونه الگوریتم QT ALOHA ..... ۳۸

**شکل ۲-۷)** نمونه الگوریتم EBFSA ..... ۴۰

**شکل ۲-۸)** نمونه الگوریتم FSAPB ، وقتی Pcoll کمتر از Pth ..... ۴۲

**شکل ۲-۹)** نمونه الگوریتم FSAPB ، وقتی Pcoll بیشتر از Pth ..... ۴۳

**شکل۳-۲-۱)** ارتباط بین PCD و PICC ..... ۵۵

**شکل۳-۲-۲)** مکث در پروتکل نوع A ..... ۵۶

**شکل۳-۲-۳)** تعیین پایان در مکث ..... ۵۷

**شکل۳-۲-۴)** حداقل محدوده کوپلینگ PICC ..... ۶۴

**شکل۳-۳-۱)** زمان تاخیر فریم PICC به PCD ..... ۷۰

**شکل ۳-۳-۲)** فریم استاندارد ..... ۷۳

**شکل ۳-۳-۳)** ساماندهی و انتقال بیت در فریم ضد تداخل بیت گرا ، FULL BYTE CASE ..... ۷۴

**شکل ۳-۳-۴)** ساماندهی و انتقال بیت در فریم ضد تداخل بیت گرا ، SPLIT BYTE CASE ..... ۷۵

**شکل ۳-۳-۵)** دیاگرام حالت های PICC نوع A ..... ۷۶

**شکل ۳-۳-۶)** فلوچارت راه اندازی و ضدتداخل برای PICC ..... ۸۱

**شکل ۳-۳-۷)** حلقه ضدتداخل ، فلوچارت برای PCD ..... ۸۵

**شکل ۳-۳-۸)** استفاده از مراحل Cascade ..... ۸۸

**شکل ۳-۴-۱)** فعالسازی PICC نوع A توسط PCD ..... ۹۴

**شکل ۳-۴-۲)** بدنه ATS ..... ۹۵

**شکل ۴-۱)** نمای شماتیک مدار پردازشگر ..... ۱۰۴

**شکل ۴-۲)** نمای PCB مدار پردازشگر ..... ۱۰۵

**شکل ۴-۳)** ماژول YLMF018 ..... ۱۰۷

**شکل ۴-۴)** ابعاد ماژول YLMF018 ..... ۱۰۸

# فهرست جداول

**جدول ۱-۱)** مقایسه RFID در باند های فرکانسی مختلف ..... ۴

**جدول۱-۲)** خصایص سیستم RFID در باند های فرکانسی مختلف ..... ۱۴

**جدول ۳-۳-۱)** زمان تاخیر فریم PICC به PCD ..... ۷۱

**جدول ۳-۳-۲)** کدگذاری فریم درخواست ..... ۷۹

**جدول ۳-۳-۳)** رمزنگاری ATQA ..... ۸۲

**جدول ۳-۳-۴)** حالت های b7 و b8 ..... ۸۲

**جدول ۳-۳-۵)** حالت های b1 تا b5 ..... ۸۳

**جدول ۳-۳-۶)** کدگذاری SEL ..... ۸۶

**جدول ۳-۳-۷)** کدگذاری NVB ..... ۸۶

**جدول ۳-۳-۸)** کدگذاری SAK ..... ۸۷

**جدول ۳-۳-۹)** اندازه ی UID ..... ۸۷

**جدول ۴-۱)** پایه های خروجی ..... ۱۰۸

**جدول ۴-۲)** ویژگی YLMF018 ..... ۱۰۹

**جدول ۴-۳)** قالب دستور ارسالی ..... ۱۰۹

**جدول ۴-۴)** قالب اطلاعات دریافتی ..... ۱۱۰

# فصل اول

# عنوان پروژه و تحقیق

**۱-۱ عنوان پروژه:**

تحقیق و طراحی سیستم های RFID ضدتداخل

**۱-۲ تعریف پروژه:**

هدف از پروژه در ابتدایکار در این فصل به بررسی سیستم های RFID به صورت کلی و جزئیات هر سیستم RFID و نحوه وقوع تداخل و همچنین به تاریخجه این سیستم ها می پردازیم . در فصل بعدی به بررسی سیستم های RFID ضدتداخلی که در جهان مورد استفاده قرار می گیرد ، می باشد . در این سیستم ها به بررسی نقاط ضعف و قوت هر پروتکل ضدتداخل می پردازیم و آنها را با یکدیگر مقایسه می کنیم . در فصل سوم به بررسی یک استاندارد مایفر موجود در ایران که پروتکل ضدتداخل خاص خود را دارد می پردازیم .

نام این استاندارد ISO/IEC 14443 می باشد که شرکت NXP مدارات خود RFID ضدتداخل را با این استاندارد روانه بازار می کند . در فصل آخر هم به یک نمونه مدار عملی که از این استاندارد استفاده می کند می پردازیم .

## ۱-۳ تاریخچهRFID

به این علت که فن‌آوری RFID اخیرا گسترش و رواج قابل ملاحظه‌ای یافته است،بسیاری از افراد تصور می‌کنند که این تکنولوژی جدید و نوست در حالی که RFID از حدود سال ۱۹۷۰وجود داشته است اما به دلیل قیمت بالا این وسیله تا سال‌های اخیر در مصارف تجاری کاربرد زیادی نداشته است. طبق بررسی‌های انجام شده مفهوم RFID از زمان جنگ جهانی دوم با کشف فن‌آوری تقریبا مشابهی به نام IFF که معرف Identify Friend or Foe می‌باشد مطرح گردیده است. IFF روشی برای تشخیص هواپیماهای جنگی دوست یا دشمن بود که توسط انگلیسی‌ها کشف و استفاده شد .IFF مکانیزمی شبیه به RFID دارد.

یک تکنولوژی مشابه دیگر در سال ۱۹۴۵ توسط "Leon Theremin "کشف شد که یک وسیله جاسوسی بود و اطلاعات صوتی را با استفاده از امواج رادیویی انتقال می‌داد.

اولین بار فن‌آوری RFIDبه شکل امروزی آن توسط "Mario Cardullo" کشف شد اما تا سال ۱۹۷۰ به علت گرانی استفاده تجاری نداشت.

## ۱-۴ اصول فن آوری RFID

RFID از سه قسمت تشکیل شده است:

- برای برقراری ارتباط و ارسال امواج رادیویی به A Scanning antenna :tag

- A Transceiver with a decoder: برای تفسیر داده‌ها
- A Transponder ( the RFID tag): که اطلاعات لازم در آن ذخیره شده است

شکل زیر یک Transceiver را نشان می‌دهد:

شکل ۱-۱ : یک Transceiver

RFID Tag خود از دو قسمت تشکیل شده: ۱)chip ۲)Antenna که در شکل زیر نمایش داده شده است:

نحوه انجام عملیات:

آنتن (Scanning Antenna) امواج رادیویی را در محدوده نسبتا کوچکی منتشرمی‌کند. این امواج رادیویی دو عمل اصلی انجام می‌دهند:

۱)وسیله‌ای برای ارتباط با RFID Tag(transponder) است.

۲)انرژی مورد نیاز tag برای برقراری ارتباط را فراهم می‌کند(در مورد tagهای passive )

وقتی که یک tag در میدان الکترومغناطیسی ایجاد شده در اطراف reader قرار می‌گیرد، سیگنال‌های فعال کننده که توسط آنتن فرستاده شده‌اند،روی آن اثر گذاشته و به عبارتی تراشه RFID را بیدار می‌کند و این تراشه اطلاعات موجود در tag را در اختیار آنتن قرار می‌دهد.نقش transceiver در این عملیات کنترل خطوط ارتباطی وداده‌ها است

در واقع یک دستگاه reader ترکیبی است از یک scanning antenna و transceiver .

اطلاعات خوانده شده توسط reader به server محلی موجود انتقال می‌یابد و این اطلاعات پردازش شده و در تشکیلات داخلی یک سازمان برای کاربردهای مختلف مورد استفاده قرار می‌گیرد. شکل زیر به صورت کلی چگونگی عملکرد RFID را توضیح می‌دهد :

انواع RFID از نظرمحدوده فرکانس:

RFID در سه محدوده فرکانس مختلف کار می‌کند:

۱- فرکانس پایین(LF): Low Frequency ←یعنی فرکانس بین ۱۲۰ تا ۱۳۴ کیلو هرتز

۲- فرکانس بالا(HF): High Frequency ← یعنی فرکانس ۱۳.۵۶ مگاهرتز

۳- فرکانس بسیار بالا(UHF): Ultra High Frequency ← یعنی فرکانس بین ۹۰۲ تا ۹۱۵ کیلو هرتز

ویژگی های این سه نوع RFID در جدول زیر مقایسه شده است:

| نوع | محدوده فرکانس | توانایی عبور سیگنال از مواد | توانایی خوانده شدن هم زمان چند tag | قیمت | کاربرد نمونه |
|---|---|---|---|---|---|
| LF | 120-134 KHz | زیاد | ضعیف | گران | شناسایی حیوانات در مراکز پرورش حیوانات یا دامداری ها |
| HF | 13.56 MHz | کم | نسبتا خوب | ارزان | کتابخانه ها |

| | | | | | |
|---|---|---|---|---|---|
| UHF | 902-915 MHz | بسیار کم | بسیار عالی | ارزانترین | حمل و نقل کالاها و موجودی ها |

جدول ۱-۱ : مقایسه RFID در باند های فرکانسی مختلف

انواع Tag های RFID:

به طور کلی سه نوع tag RFID وجود دارد که عبارتند از:

۱- tag های Passive: این نوع tag ها هیچ منبع تولید انرژی درونی ندارند و انرژی خود را از طریق سیگنال‌های RF که توسط دستگاه Reader ارسال و توسط آنتن موجود در tag دریافت می‌شود، تامین می‌کنند.

۲- tag های Semi-passive: بسیار شبیه tag های Passive است ؛ با این تفاوت که باتری کوچکی در آن‌ها وجود دارد و انرژی لازم برای فعال شدن مدار داخل آن‌ها را فراهم می‌سازد.

۳- tag های Active: این tag ها دارای یک منبع انرژی داخلی می‌باشند که توانایی انتقال اطلاعات در فواصل دورتر را فراهم می‌کند.

این سه نوع tag از جهات دیگری چون سایز، دامنه پاسخ‌گویی، سرعت پاسخ‌گویی و... نیز با هم تفاوت هایی دارند .

این خواص با حرکت از tag های Passive به سوی Active به صورت زیر تغییر می‌کنند:

اندازه : افزایش

*اندازه کوچک‌ترین tagPassive ← 0.15mm× 0.15mm

*اندازه کوچک‌ترین Active tag ← به اندازه یک سکه

دامنه(Range) پاسخ‌گویی : افزایش

قیمت : افزایش

سرعت پاسخ‌گویی : افزایش

قابلیت اطمینان : افزایش

عمر این برچسب‌ها به صورت زیر تغییر می‌کند:

Passive > Active > Semi-passive

*دلایل تفاوت عمرtag ها:

۱- tag هایی که منبع انرژی داخلی دارند به علتمحدودیت منبع عمر محدود دارند.

۲- باتری کوچک موجود در برچسب Semi-passive عمر کوتاه تری از منبع انرژی برچسب Active دارد.

مقایسه RFID با بارکدو مزایای عمده RFID:

۱- دستگاه‌های خواننده بارکدها در صورتی عمل می‌کنند که برچسب در مسیر خط مستقیم دید آن‌ها قرار گیرد؛ در حالی که هیچ یکاز انواع RFID tag برای خوانده شدن احتیاجی به قرار گرفتن در مسیر دید مستقیم Reader ندارند.

۲- Tag های RFID می‌توانند از فاصله بسیار دورتری نسبت به بارکد خوانده شوند.یک RFID Reader می‌تواند اطلاعات RFID را تا فاصله ۳۰۰ فوت هم بخواند،در حالی که فاصله خوانده شدن بارکد بسیار کمتر است و عملا بیشتر از ۱۵ فوت نیست.

۳- چند RFID tag می‌توانند به طور هم‌زمان خوانده شوند اما بارکد این مزیت را ندارد.

۴- خواندن اطلاعات از RFID با سرعت بسیار بالاتری صورت می‌گیرد.(حدود ۴۰ عدد یا بیشتر در ۱ ثانیه)؛ در حالی که خواندن اطلاعات از بارکد بسیار زمان‌برتر است.به علت این که tag باید دقیقا روبه روی Reader قرار گیرد، ممکن است چند ثانیه برای خواندن فقط یکtag وقت صرف شود.

۵- از آن جایی که بارکد باید برای خوانده شدن در مسیر دید مستقیم gun قرار گیرد، باید حتما بر روی سطح خارجی کالا نصب شود.در نتیجه خیلی سریع آسیب دیده و غیر قابل استفاده می‌شود.در حالی که می‌توان برچسب‌های RFID را در داخل پوشش پلاستیکی قرار داد وحتی می‌توان آن‌ها را در داخل محصول فرو کرد که این خود، دوام آن‌ها و امکان استفاده مجدد از آن‌ها را فراهم می-سازد.

۶- عمر برچسب‌های RFID از بارکد بیشتر است.

۷- RFID توانایی کار در محیط‌های خشن را داردو کارایی آن بسیار بیشتر از بارکد است.

۸- بارکدها برچسب‌هایی فقط خواندنی هستند و اطلاعات آن‌ها قابل تغییر نمی‌باشد.درحالی که RFID توانایی خوانده شدن و نوشته شدن مجدد را دارد.Reader ها قادرند با برچسب‌ها ارتباط برقرار کنند و تا جایی که طراحی برچسب اجازه می‌دهد اطلاعات آن را تغییر دهند.بنابراین توسط بارکد تنها می‌توان یک کالا را رد گیری نمود.اما با استفاده از RFID می‌توان عملیاتی چون ثبت وقایع، پارامترها و اندازه‌گیری‌ها را نیز اجرا کرد.

۹- برچسب‌های RFID گران‌تر ازبارکد هستند و این قیمت بالا گاهی اوقات به حدی تاثیر گذار است که صرف نظر از تمام مشکلات بارکدها استفاده از آن‌ها بسیار به صرفه‌تر از به‌کارگیری RFID می-باشد.

۱۰- امنیت اطلاعات موجود در روی RFID کمتر از بارکد است.زیرا دستگاه‌های Reader مختلف از فواصل دور قادر به خواندن این اطلاعات می‌باشند.

۱۱- تعداد بایت های موجود برای ذخیره‌سازی اطلاعات در RFID بسیار بیشتر از بارکد است.بنابراین می‌توان در یک برچسب RFID اطلاعات فراوانی از جمله : کد کالا، محل ذخیره و نگه‌داری، محل

تولید، تاریخ مصرف، قطعات و مواد تشکیل دهنده، حمل و نقل‌های صورت گرفته و بسیاری اطلاعات دیگر را ذخیره نمود.

کاربردهای RFID:

RFID در جاهای مختلفی از پزشکی تا صنعت و حتی در موجودات زنده کاربرد دارد

مثال‌هایی از این کاربردها در ادامه آورده شده است:

۱)مدیریت زنجیره تامین(SCM):

با قرار دادن tagهای RFID بر روی اجناس در زنجیره تامین در هر لحظه می‌توان تشخیص داد که کالای مورد نظر در چه مرحله ای از زنجیره قرار دارد.RFID توسط شرکت‌های مختلف در دنیا از جمله wal-mart، Gillete و... برای این منظور به کار برده شده است.

کمک عمده RFID در زنجیره تامین به هنگام خرید مواد اولیه از تامین کنندگان و همچنین انتقال محصولات به توزیع کنندگان می‌باشد. Tagهای RFID که بر روی پالت‌های حمل محصولات نصب می-شوند اطلاعات کاملی از محصولات موجود در پالت در اختیار مراکز بازرسی قرار می‌دهد و در نتیجه احتیاجی به باز کردن محموله و شمارش دستی محصولات نیست که این امر سبب کاهش قابل توجهی در هزینه بازرسی و افزایش دقت و کاهش اشتباهات حمل خواهد شد . همچنین اطلاعات وسیعی درباره مواد و قطعات تشکیل دهنده محصول ،مراحل ساخت آن ، زمان ساخت و تحویل ،محل قرارگیری آن در انبار و... را می‌توان در داخل tagها ذخیره و نگهداری کرد.

۲)سیستم کنترل موجودی :

یکی از کاربردهای بسیار متداولRFID کاربرد آن در برنامه ریزی و کنترل موجودی‌هاست. اطلاعات موجودی‌ها در tagهای RFID نگهداری می‌شود و از این طریق هر لحظه می‌توان میزان موجودی موسسه و محل نگهداری آن‌ها را چک کرد و احتیاجات را مشخص و سفارشات لازم را ارسال نمود. RFID هزینه کنترل موجودی را کاهش داده و کارایی ودقت آن را به میزان قابل توجهی افزایش می-دهد.شرکتwal-mart ادعا می‌کند که با استفاده ازRFID هزینه کنترل موجودی شرکت به میزان ۱/۵ ٪ کاهش داشته است.

*شرکتwal-mart در زمینه استفاده ازRFID تقریبا پیشگام است. با استفاده ازاین سیستم، wal-mart در هر لحظه چک می‌کند که از هر محصول چقدر و در کدام قفسه‌ها وجود دارد و اگر در هر جا مقدار این کالاها کم باشد به طور خودکار دستوری به قسمت انبار داری ارسال می‌شود تا قفسه‌ها مجددا پر شوند. همچنین در زمان برداشتن یا گذاشتن موجودی در انبار این اطلاعات ثبت شده،سطح موجودی انبار چک شده و در صورت نیاز سفارش به تامین کنندگان به صورت خودکار داده می‌شود.

۳)ایجاد امنیت و جلوگیری از سرقت :

نمونه چنین کاربردی در فروشگاه‌های زنجیره‌ای ، پوشاک ،کتابخانه‌ها و... دیده می‌شود. tagهای متصل بر روی اجناس باید در هنگام خرید و پرداخت پول توسط مشتری غیر فعال شوند در غیر این صورت gateهای کار گذاشته شده نزدیک در ورودی فروشگاه‌ها در هنگام خروج، tag فعال را شناسایی و سیستم امنیت را به کار می‌اندازد.مثال دیگری از این کاربرد در جلوگیری از سرقت اتومبیل است ،برچسب RFID در داخل کلید قرار می‌گیرد و تا زمانی که کلید مورد نظر، نباشد ماشین حرکت نمی‌کند.

کاربرد عمده RFID امروزه در کتابخانه‌ها می‌باشد.

۴)پزشکی:

در بیمارستان‌ها این tagها برای تعیین موقعیت تجهیزات و پرسنل در هر لحظه به کار می‌روند.علاوه بر آن برای کنترل موجودی‌ها از جمله خون،دارو و... نیز کاربرد دارد. همچنین کاربردجدیدی از RFID برای انسان‌ها وجود دارد که در بخش کاربردهای آینده بیشتر در مورد آن بحث می‌شود.

۵) کنترل و نگهداری موجودات زنده:

در مراکز نگهداری از حیوانات یا دامداری‌ها کاربردهای فراوان RFID قابل مشاهده است. با جاسازی یک برچسب RFID در زیر پوست حیوانات می‌توان کد منحصر به فردی به هر حیوان تخصیص داد و در نتیجه از این طریق در هر لحظه می‌توان موقعیت حیوان را مشخص کرد ،همچنین می‌توان کنترل کرد که آیا به هرحیوان غذا، داروهای مورد نیاز،مراقبت کافی و... داده شده‌است یا خیر. بنابراین از این طریق بسیاری از هزینه‌هایی که به علت اشتباهات پزشکی و مراقبتی ایجاد می‌شود، کاهش می‌یابد.

علاوه بر این مالکان و دامپزشکان می‌توانند آمار دقیق حیوانات تحت مسئولیت خود را بدانند.کاربرد RFID در انتقال حیوانات هم بسیار مهم است. به عنوان مثال در گاوداری‌ها شیوع بیماری‌هایی چون جنون گاوی این ضرورت را فراهم می‌کند که درباره هر حیوان بدانیم چه غذاها و داروهایی مصرف نموده و در چه محل-هایی نگهداری شده‌است. RFID برای شناسایی و مراقبت از حیوانات خانگی نیز به کار می‌رود. به عنوان مثال اگر یک حیوان خانگی گم شود و آن را به مراکز مربوطه تحویل دهند، آن‌ها قادرند با خواندن tagRFID اطلاعاتی در مورد صاحب آن به دست آورند.

۶) در محل های فروش : Point-of-sale (POS)

Tag ها برای خرید و فروش‌های غیر نقدی مورد استفاده قرار می‌گیرند. مانند ایستگاه‌های سوخت، ایستگاه مترو و...

مثلا در یک ایستگاه مترو یک دستگاه Reader در قسمت بالای در ورودی نصب می‌شود که با عبور مسافران از این در اطلاعات موجود در برچسب RFID روی کارت آن‌ها خوانده شده و به صورت خودبه خود از حساب Credit-card فرد، پول بلیت دریافت و به او اجازه عبور داده می‌شود؛ که این کار هزینه صدور بلیت و چک کردن بلیت‌ها و احتمال اشتباه را کاهش می‌دهد و زمان انتظار مسافران کمتر خواهد بود.

۷) مدیریت منابع انسانی (HRM) :

از طریق تخصیص یک برچسب RFID به هر یک از کارکنان سازمان می‌توان اطلاعات موردنیاز درباره آن فرد را با کد اختصاصی خودش ذخیره و در مواقع لزوم به آن دسترسی پیدا کرد؛ همچنین با قرار دادن دستگاه‌های Reader در محل‌های مختلف در کارخانه می‌توان عبورو مرور و زمان ورود و خروج کارکنان را کنترل نمود.

۸) کنترل مسافران:

یکی از کاربردهای مهم RFID در کنترل مسافران مثلا از طریق وجود یک برچسب RFID بر روی pass-port مسافران می‌باشد که در هنگام عبور از gateهای ورودی به سالن پرواز و... اطلاعات آن خوانده شده و به مسافر اجازه عبور داده می‌شود. بنابراین کنترل افراد ازاین راه بسیار مطمئن‌تر، کم‌هزینه‌تر و سریع‌تر خواهد بود. اطلاعاتی از جمله زمان و مکان کلیه مسافرت‌ها و ورود و خروج از کشورها بر روی این tagها ذخیره می‌شود.

۹) دریافت عوارض راه:

در بسیاری از کشورها برروی ماشین‌ها برچسب‌هایی با اطلاعات شماره credit-card و مشخصات مالک آن، قرار داده می‌شود.

در هنگام عبور از جاده‌ها gateهایی برای جمع‌آوری عوارض وجود دارد که این tagها را در حال حرکت خوانده و به صورت خودکار، مقدار عوارض را از حساب صاحب اتومبیل کم می‌کند. از این طریق توقف برای پرداخت عوارض در جاده‌ها حذف خواهد شد.

## کاربردهای آینده RFID:

### ۱)ایجاد سیستم‌های هوشمند خانگی:

به عنوان مثال: ساخت یک ماشین لباس‌شویی که قادر است tagهای RFID روی لباس‌ها را خوانده و با توجه به اطلاعات آن‌ها از جمله نوع پارچه لباس، رنگ و... درجه شست و شو را تنظیم کند و در صورتی که فرد قصد شستن ترکیب نادرستی از رنگ و جنس لباس‌ها را دارد، اخطار لازم را به او بدهد.

مثال دیگر یخچالی است که می‌تواند مواد غذایی داخل خود را شناسایی نموده و هر زمان که یکی از آن-ها(مثلا شیر) کم شد یا تاریخ انقضای آن نزدیک شد به صاحب‌خانه خبر بدهد؛ یا حتی قادر است یک لیست خرید آماده کند!

### ۲)ایجاد سیستم‌های اعلام خطر در اتومبیل:

مثلا استفاده از یک برچسب هوشمند RFID در تایر اتومبیل که وقتی باد لاستیک کم می‌شود یا مشکلی به وجود می‌آید، به راننده هشدار داده و از بروز تصادف جلوگیری می‌کند.

### ۳) در مصارف پزشکی:

طرح آینده برای کاربردهای پزشکی که در بعضی مکان‌ها نیز در شرف اجراست، این است که برای هر کودکی که به دنیا می‌آید یک برچسب RFID طراحی و در بدن او جاسازی شود.(این برچسب معمولا در زیر پوست، با یک روش ساده جاسازی می‌شود.)

تمامی اطلاعات پزشکی فرد از جمله: گروه خونی، ویژگی‌ها(قد، وزن و...)، سوابق پزشکی و... بر روی آن برچسب نگهداری شده و هر بار که فرد به پزشک مراجعه می‌کند، پزشکان قادرند تمامی اطلاعات مورد نیاز خود را برای تشکیل پرونده و تشخیص وضعیت بیمار خود با اسکن کردن برچسب RFID آن‌ها به دست آورند.

در حال حاضربرچسب‌هایی برای این منظور ساخته شده‌اند که هیچ مزاحمت و حساسیتی برای بافت‌های بدن ایجاد نمی‌کند و از نظر اندازه، به اندازه یک دانه برنج است.

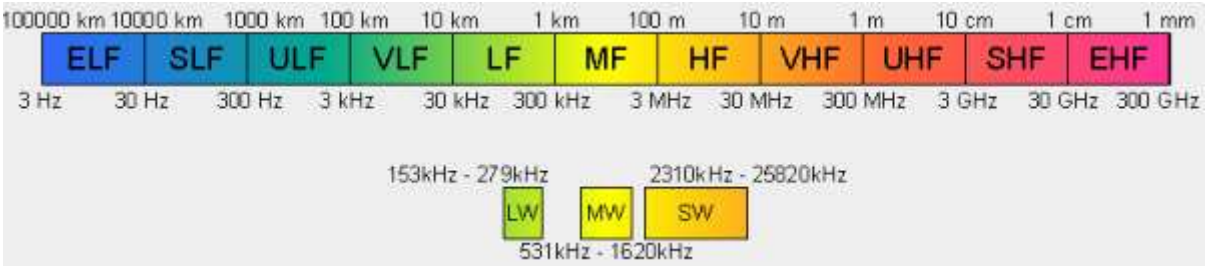
تنها مشکلی که در این زمینه وجود دارد امکان جابه جایی tag در داخل بدن است که این هم از طریق تزریق موادی به بافت‌های اطراف tag که باعث رشد این بافت‌ها و احاطه شدن tag توسط آن‌ها می‌شود، امکان پذیر است.

از آن‌جا که این tagها به گونه‌ای ساخته شده‌اند که فقط از فاصله بسیار نزدیک خوانده می‌شوند،در نتیجه باید بسیار نزدیک به سطح پوست قرار داده شوند.

**اجزاء یک سیستم RFID : طیف فرکانس**

یکی از ملاحظات مهم در ارتباط با فناوری RFID ، فرکانس عملیاتی آنان است . همانند تلویزیون که می تواند در باندهای UHF ( برگرفتهشدهاز HighFrequency-Ultra ) و VHF ( برگرفته شده از Very High Frequency ) فعالیت نماید ،سیستمهایRFID نیز می توانند از باندهای مختلفی برای ارتباطات خود استفاده نمایند .

شکل ۱ طیف فرکانس رادیویی را نشان می دهد .




شکل ۱-۲ : طیف فرکانس رادیویی

در RFID هم از باند فرکانس پائین و هم از باند فرکانس بالا در محدوده های زیر استفاده میگردد .

- **باندهای فرکانس پایین RFID**

- فرکانس پایین و یا LF ( برگرفته شده از LowFrequency ) : بین ۱۲۵ تا ۱۳۴ کیلوهرتز

- فرکانسبالا ویاHF ( برگرفته شده از HighFrequency ) : محدوده ۵۶ / ۱۳ مگاهرتز

- **باندهای فرکانس بالا RFID**

- باندUHF: محدوده ۸۶۰ تا ۹۶۰ مگاهرتز

- میکروویو : ۵ / ۲ گیگاهرتز به بالا

انتخاب فرکانس بر روی چندین خصلت سیستم های RFID تاثیر گذار است که در ادامه به تشریح برخی از آنان خواهیم پرداخت .

**اندازه و قیمت تگ های RFID**

سیستم های RFID اولیه عموما" از باند LF استفاده می کردند ( به دلیل سهولت در ساخت ) . این نوع سیستم ها دارای مشکلات مختص به خود می باشند نظیر ابعاد بزرگ آنتن که می تواند قیمت تمام شده آنان را افزایش دهد .

در حال حاضر استفاده از باند HF بسیار متداول است .با توجه به پیشرفت های اخیر در فناوری ساخت

تراشه ها ، قیمت تگ های UHF قابل رقابت با تگ های HF شده است . تگ های RFID میکروویو مشابه تگ های UHF می باشند با این تفاوت که می توان آنان را کوچکتر و با قیمت کمتری تولید کرد.

جدول ۱-۲ : خصایص سیستم RFID در باندهای فرکانسی مختلف

| باند فرکانس | LF 125 KHZ | HF 13.56MHZ | UHF 860-960 MHZ | Microwave 2.5 GHZ به بالا |
|---|---|---|---|---|
| محدوده خواندن ( تگ های غیرفعال ) | کمتر از 62 سانتی متر | کمتر از یک متر | بین سه تا نه متر | چندین متر |
| منبع تامین انرژی تگ | عموما" غیرفعال می باشند | عموما" غیرفعال می باشند | عموما" فعال ولی ممکن است به صورت غیرفعال نیز باشند | عموما" فعال ولی ممکن است به صورت غیرفعال نیز باشند |
| قیمت تگ | نسبتا" گران | گران ولی بمراتب کمتر از LF | دارای پتانسیل لازم جهت تولید ارزان | دارای پتانسیل لازم جهت تولید ارزان |
| کاربردهای عمومی | ورود بدون کلید به مکان هایی خاص نظیر اطاق ها و ...<br>ردیابی حیوانات<br>پیشگیری از سرقت اتومبیل از طریق خاموش کردن آن از را دور | کارت های هوشمند<br>ردیابی آیتم هایی نظیر حمل بار<br>کتابخانه ها | ردیابی حمل بار<br>جمع آوری الکترونیکی عوارض<br>حمل بار مسافر | جمع آوری الکترونیکی عوارض |
| نرخ داده | کند | متوسط | سریع | بسیار سریع |
| کارآیی پس از قرار گرفتن در مجاورت فلزات و مایعات | خیلی خوب | خوب | بد | بدتر |
| اندازه تگ غیرفعال | خیلی بزرگ | بزرگ | کوچک | کوچک تر |

# فصل دوم

# سیستم های RFID ضدتداخل

## ۲-۱ مقدمه

در این فصل به بررسی انواع پروتکل های ضدتداخل رایج در دنیا می پردازیم . همجنین به تحقیقاتی که هم اکنون در سرتاسر جهان در این زمینه انجام می شود نگاهی می اندازیم و پروتکل ها را با یکدیگر مقایسه خواهیم کرد .

## ۲-۲ معرفی سیستم های RFID ضد تداخل

سیستم های RFID امروزه به طور گسترده ای در تشخیص خودکار مورد استفاده قرار می گیرد . هر شئ می تواند به عنوان یک کد الکترونیکی شناخته شود . در ابتدا هدف اصلی استفاده از تگ RFIDبرای جانشینی سیستم های بارکد قدیمی بود . علاوه بر اینکه تگ های RFID به خوانده شدن توسط باریکه ی نور احتیاج ندارند , این تگ ها در مقابل آب و کثیفی نیز مقاوم هستند . در ضمن , تگ ها می توانند دارای حافظه ی خواندن و نوشتنی نیز باشند که قابلیت نگهداری اطلاعات بیشتری نسبت به بارکد می باشند . این خصوصیات علت اصلی هجوم شرکت ها و کارخانجات برای استفاده از این سیستم می شود .

یک تگ RFID شامل دو بخش اصلی است : یک آی سی برای ذخیره اطلاعات و انجام عملیات مخابراتی و یک آنتن متصل به آن جهت دریافت و ارسال سیگنال رادیویی . انواع گوناگونی از تگ ها وجود دارند که بنابر منبع تغذیه و روش ارتباط آنها با سیستم تقسیم بندی می شوند . تگ های غیر فعال عموما منبع تغذیه ی داخلی ندارند و انرژی خود را از سیگنال دریافت شده تهیه می کنند . از سوی دیگر تگ های فعال یک باتری جهت تامین انرژی خود دارند که می توانند علاوه بر ارتباط با Reader با تگهای دیگر هم ارتباط بر قرار کنند . یک تگ نیمه غیر فعال ترکیبی از دو نوع ذکر شده می باشد به طوری که انرژی برای پردازش اطلاعات را از باطری خود تهیه میکند اما جهت ارتباط با Reader از روش غیر فعال استفاده می کند .

در یک سیستم RFID یک Reader توانایی ارتباط با چندین تگ که در محدوده ی آن باشند را دارد . اما روند شناسایی تگ ها ممکن است با مشکل مواجه شود اگر چندین تگ به طور همزمان اطلاعات خود را برای Reader ارسال کنند . سیگنال های ارسالی توسط تگ ها ممکن است با یکدیگر تداخل کند , بنابراین Reader ممکن است اطلاعات درستی دریافت نکند . اگر چنین چیزی اتفاق بیفتد , تگها مجبور به ارسال مجدد اطلاعات خود می شوند که باعث اتلاف زمان خواندن و کم شدن کارایی سیستم می شود . چنین مشکلی را " تداخل تگ ها " در سیستم RFID می نامند .

برای حل مشکل تداخل تگ ها , محققان هنوز به دنبال موثرترین راه برای سیستمی ضد تداخل که سریع ترین شناسایی که دقت شناسایی اطلاعات نزدیک به ۱۰۰٪ را داشته باشد , می باشند . مشکل تداخل معمولا به دو نوع تداخل در Reader و تداخل در تگ ها تقسیم بندی می شود .

پروتکل هایی که امروزه برای حل مشکل تداخل تگها استفاده می شوند هر کدام دارای معایب و مزایای مخصوص به خود می باشند . در ادامه نگاهی مختصر به این پروتکل ها می اندازیم و معایب و مزایای هر کدام را ذکر می کنیم . از آنجایی که امروزه بیشتر از تگ های غیر فعال به دلیل هزینه و حجم بسیار کم استفاده می شود , ما نیز به تفسیر روش های ضد تداخل اینگونه از تگها می پردازیم .

## ۲-۳ پروتکل های ضد تداخل برای سیستم های RFID غیرفعال

تکنولوژی های دسترسی چندگانه امروزه به صورت وسیع در مخابرات مدرن استفاده می شوند که این تکنولوژی ها به سیستم های مخابراتی اجازه می دهند که از کانال ارتباطی به طور همزمان با کمترین تداخل با یکدیگر استفاده کنند . در یک سیستم Reader , RFID معمولا با تگهایی که در محدوده اش باشند ارتباط برقرار میکند , که همه ی این ارتباطات از یک کانال ارتباطی که همان هوای اطراف است استفاده می کند . به همین دلیل از تکنولوژی دسترسی چندگانه برای سیستم های RFID استفاده می شود . از قبیل این تکنولوژی ها می توان به زمان بندی دسترسی چندگانه (TDMA) , رمزگذاری دسترسی چندگانه (CDMA) , مکان یابی دسترسی چندگانه (SDMA) و فرکانس بندی دسترسی چندگانه (FDMA) و همچنین تکنولوژی های ترکیبی اشاره کرد که در ادامه به توضیح مختصری راجع به هر کدام می پردازیم .

## ۲-۳-۱ زمان بندی دسترسی چندگانه (TDMA)

با این تکنولوژی Reader و تگها میتوانند برای ارتباط از یک فرکانس در محدوده Reader استفاده کنند و پاسخ هر تگ توسط فاصله ی زمانی که استفاده می کند شناسایی می شود . این نوع تکنولوژی محبوب ترین نوع در سیستم های RFID ضد تداخل می باشد که به راحتی می تواند با تکنولوژی های دیگر ترکیب شود . تکنولوژی TDMA به دو دسته تقسیم می شود : طرحهای احتمالی (Probabilistic) و طرحهای قطعی (Deterministic) . طرحهای قطعی معمولا به طرح های جستجوی درختی دو دویی ( Binary tree-search) نامیده می شوند . در این طرح ها همه ی تگ ها به طور قطع خوانده خواهند شد به دلیل اینکه هر شاخه منجر به برگ نشان دهنده ی یک تگ می باشد و همه ی تگها با جستجوی همه ی شاخه ها شناسایی خواهند شد .در سمت دیگر , طرح های احتمالی به طرح های مبتنی بر slotted ALOHA نامیده می شوند . در این طرح , هر تگ شانس با موفقیت شناسایی شدن را خواهد داشت , اگر چه امکان اینکه چندین تگ به خاطر تداخل شناسایی نشوند هم وجود دارد . غیر از دو نوع گفته شده طرح های ترکیبی نیز ارائه شده است .

به دلیل اینکه جستجوی درختی دودویی یک شاخه را به طور کامل برای بدست آوردن شماره تگ بررسی می کند , این حالت به طور طبیعی کند می باشد و هر چقدر تعداد تگ ها زیاد تر شود زمان بازیابی همه تگ ها افزایش پیدا می کند . برای کاهش این مشکل طرح های مختلفی ارائه شده است . طرح های درختی نیز به دو دسته اصلی تقسیم می شوند : Binary Tree Walkingو Query Tree .

در مقایسه با طرح های جستجوی درختی , طرح slot ALOHA معمولا تگ ها را سریعتر بررسی می کند . در این طرح راه ارتباطی بین Reader و تگ به شکاف های زمانی زیادی بخش بخش می شود . هر شکاف زمانی می تواند خالی باشد ( به این معنی که هیچ تگی در این شکاف زمانی پاسخ دهد ) , تداخل داشته باشد ( یعنی چند تگ در این شکاف اطلاعات خود را ارسال کند ) , یا با موفقیت تمام فقط یک تگ اطلاعات را ارسال کند که در نتیجه اطلاعات آن بازیابی می شود . کارایی این طرح به احتمال دریافت موفق در هر شکاف زمانی مربوط می شود . واضح است که نسبت هر شکاف به تعداد تگهای موجود در محدوده Reader بازده سیستم را مشخص می کند . زمانی که تعداد تگهایی ناشناس توسط سیستم از قبل مشخص نباشد , انتخاب اندازه شکاف مناسب یک عامل مهم برای تعیین کارایی یک سیستم می باشد . همچنین اندازه مناسب شکاف معمولا متغیر است به دلیل اینکه تعداد تگ های خوانده نشده در طول زمان کاهش می یابد , که یک چالش برای انتخاب مناسب ترین اندازه شکاف را ایجاد می کند .

## ۲-۳-۲ کدبندی دسترسی چندگانه CDMA

همانطور که پیشتر اشاره شد , طرح های ضد تداخل مبتنی بر TDMA می توانند فقط یک شماره تگ را در یک شکاف زمانی بازیابی کند . برای افزایش بازده سیستم همراهی تکنولوژی TDMA با تکنولوژی های دیگر ضروری است .

تکنولوژی CDMA به طور وسیعی در بسیاری از سیستم های مخابرات مدرن استفاده می شود .طرح های CDMA به چندین ارتباط اجازه می دهد که در کنار یکدیگر در یک فرکانس و یک زمان حضور داشته باشند .تکنولوژی CDMA معمولا به دو گروه تقسیم بندی می شود : frequency hopping CDMA (FH-CDMA) و direct sequence spreading CDMA (DS-CDMA) . به دلیل اینکه یک تگ غیر فعال نمی تواند فرکانس ارتباطی خود را انتخاب کند و فقط می تواند از سیگنال دریافتی که Reader منتشر کرده استفاده کند , FH-CDMA بزای این نوع تگ ها مناسب نمی باشد . اما تکنولوژی DS-CDMA از سیگنال پخش شده به عنوان نشان های از سیگنال منبع استفاده می کند . منابع سیگنال مختلفی می توانند در همزمان سیگنال های خود را ارسال کنند و Receiver می تواند سیگنال های دریافتی را با توجه به نشانه اش جدا کند .

یک تگ غیر فعال می تواند سیگنال مورد نظر را تغییر ضریب انعکاس آنتن خود برای تولید سیگنال مناسب برگشتی ایجاد کند. با استفاده از این کار تگ می تواند به راحتی سیگنال کد شده ی دلخواه را تولید کند .

## ۲-۳-۳ مکان یابی دسترسی چندگانه SDMA

SDMA نسسبتا تکنولوژی جدیدتری در سیستم های مخابرات مدرن در مقایسه با سه تکنولوژی دیگر می باشد . ساختار SDMA قابلیت تامین یک دسترسی بدون تداخل در کانال بی سیم را دارا می باشد . به این دلیل که در این روش می توان چندین کانال را در حوزه مکانی ایجاد کرد . با این کار ظرفیت کانال در یک فرکانس و یک شکاف زمانی به طور قابل ملاحظه ای افزایش می یابد .

عموما در یک طرح SDMA کانال های انتقال توسط زاویه ی دریافت از یکدیگر جدا می شوند . در سیستم RFID غیر فعال , این تکنیک بسیار سودمند است به خاطر اینکه به هیچ تغییر فیزیکی در پروتکل ارتباط نیاز ندارد و فقط به آنتن های جهتی نیاز دارد .

مانند روش های CDMA , یک سیستم یکپارچه که از ترکیب TDMA و SDMA تشکیل شود هنوز طرح نشده است . برای مثال اگر تگ ها به طور یکدست در زیر شاخه ها پخش نباشند , اختصاص شکاف زمانی باید به گونه ای که بازده سیستم بهینه باشد طراحی شود .

### ۲-۳-۴ فرکانس بندی دسترسی چندگانهFDMA

تکنولوژی FDMA به چندین کانال انتقال اجازه ی کار در یک زمان با فرکانس های کار مختلف را می دهد . در یک سیستم RFID غیر فعال , سیگنال توسط Reader در چند فرکانس منتشر می شود تا توان و دستور Reader را به تگ های غیر فعال منتقل کند .یک تگ نه تنها از سیگنال به عنوان تغذیه ی خود استفاده می کند , بلکه از این سیگنال به عنوان سیگنال کریر مدولاسیون برگشتی نیز بهره می برد .بنابراین در این تکنولوژی می توان با جدا کردن فرکانس های مختلف چندین تگ را همزمان بازیابی کرد .

همراهی این تکنولوژی با slotted ALOHA , بازده سیستم را تقریبا دو برابر می کند زیرا شکاف های زمانی بی کار می تواند به طور کامل استفاده شود . اما سیستم هنوز اجازه ی بازیابی یک شماره تگ در یک شکاف زمانی را می دهد . یکی دیگر از معایب این طرح , مسئله ی هزینه می باشد که انتظار می رود با رشد این تکنولوژی به مرور زمان کاهش یابد .

## ۲-۴ بررسی کامل پروتکل TDMA

الگوریتم های ضد تداخل تگ ها می توانند به دو گروه تقسیم شوند : روش های احتمالی و روش های قطعی . الگوریتم های احتمالی مانند ALOHA , Slotted ALOHA و Frame Slotted ALOHA احتمال رخ دادنتداخل تگها را با اجازه دادن به تگ ها برای ارسال شماره سریالشان در زمانمجزا کاهش می دهند . در ALOHA تگها به صورت تصادفی زمان ارسال خود را انتخاب می کنند و در Slotted ALOHA تگ ها فقط در آغاز یک برش زمانی ارسال می کنند که در دوره ی زمانی مشخص است . Frame

Slotted ALOHA که بهترین انتخاب از الگوریتم های احتمالی است , یک فریم را با تعداد زیادی برش زمانیش پیکربندی می کند . هنگامی که یک تگ ID خود را فقط در یک برش زمانی در هر فریم ارسال می کند . FS ALOHA تداخل را کاهش می دهد . هر چند الگوریتم های احتمالی نمی توانند به طور کامل از تداخل جلوگیری کنند . علاوه بر این اگر تعداد تگ ها مشخص نباشد , این الگوریتم ها نمی توانند تضمین کنند که همه ی تگ ها در یک دوره ی زمانی مشخص بازیابی شوند. بنابراین یک تگ مشخص ممکن است برای مدت زمان طولانی شناسایی نشود و سبب مشکلی شود که مشکل tag starvation نامیده می شود . برای الگوریتم های قطعی , ریدر یک دستور درخواست می فرستد که بر اساس ID های تگ ها است تا تگهای در ناحیه بازپرسی پاسخ دهند .

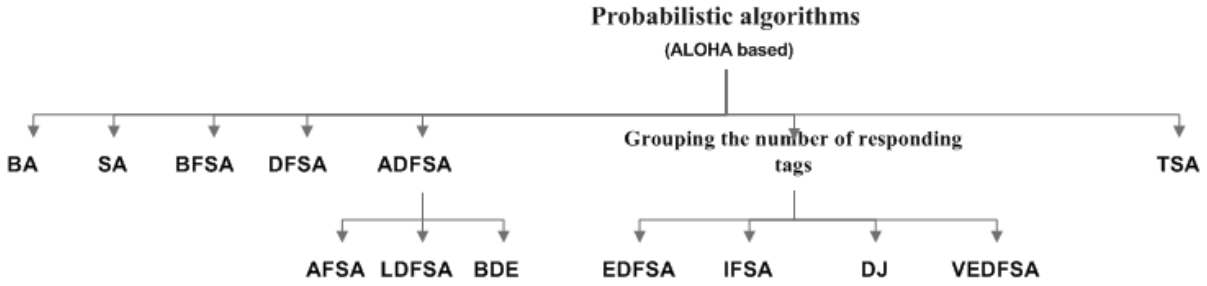
این روش ها بر اساس الگورتم های ضدتداخل درختی مانند : Binary Tree و Query Tree هستند . در الگوریتم های باینری اگر تداخل رخدهد , ریدر گروه تگهای مختل شده را به دو گروه تقسیم می کند تا یک تگ بدونتداخل شناسایی شود . الگوریتم های Binary Tree روش های مختلفی دارد . مثلا سرشماری یک لیست از تگها یا ایجاد برخی تغییرات در الگوریتم جستجوی باینری . در الگوریتم های Query Tree ریدر در هر تکرار ارتباطی یک پیشوند را ارسال می کند و تگ ها در صورت هماهنگ بودن این پیشوند با قسمتی از ID آنها , ID خود را باز می گردانند . اگر برای یک پیشوند خاص تداخل روی دهد ریدر پاسخ را رد می کند و پیشوند را با یک بیت توسعه می دهد . بازده فرایند شناسایی برای الگوریتم های قطعی می تواند توسط طول و توزیع ID های تگ و تعدادتگ ها تحت تاثیر قرار گیرد .

هر چند این روش ها تداخل را کاهش داده و در مواردی مشکل را حل کرده , اما این روش ها در کاربرد های عملی به اندازه کافی سریع و پر بازده نمی باشند . از آنجایی که الگریتم های ضد تداخل تگ بر اساس Tree نرخ بازیابی کامل را بدست می دهد و الگوریتم های ضد تداخل بر اساس SALOHA پیاده سازی ساده و کارایی خوبی برای تعداد کمی از تگ ها را دارن , برخی تحقیقات بر روی ترکیب این دو الگوریتم تمرکز کرده اند . بنابراین یک روش جدید الگوریتمهای ضد تداخل ایجاد شده که الگوریتمهای Tree –

ALOHA هستند . این الگوریتم ها را می توان الگوریتم های قطعی با کارایی بالا به منظور پروسه ALOHA درنظر گرفت .

### ۲-۴-۱ الگوریتم های ضد تداخل بر اساس ALOHA

تمام الگوریتم های احتمالی اساسا فرم های اصلاح شده ی الگوریتم ALOHA می باشند . این الگوریتم ها احتمال بروز تداخل تگ را کاهش می دهند , زیرا تگ ها سعی می کنند در زمانهای مجزا پاسخ دهند .




BA=Basic ALOHA Algorithm
SA=Slotted ALOHA Algorithm
BFSA= Basic Framed Slotted ALOHA Algorithm
DFSA= Dynamic Framed Slotted ALOHA Algorithm
ADFSA= Advanced Dynamic Framed Slotted ALOHA Algorithm
AFSA= Advanced Framed Slotted ALOHA Algorithm
LDFSA= Learning-Based Dynamic Framed Slotted ALOHA Algorithm
BDE= Binomial Distributing Estimation Algorithm
EDFSA= Enhanced Dynamic Framed Slotted ALOHA Algorithm
IFSA= Improved Framed Slotted ALOHA Algorithm
DJ= Detection and Jump algorithm
VEDFSA= Variant Enhanced Dynamic Frame Slotted ALOHA Algorithm
TSA= Tree Slotted ALOHA Algorithm

شکل ۲-۱ : الگوریتم های احتمالی

### ۲-۴-۲ الگوریتم ALOHA پایه و Slotted ALOHA

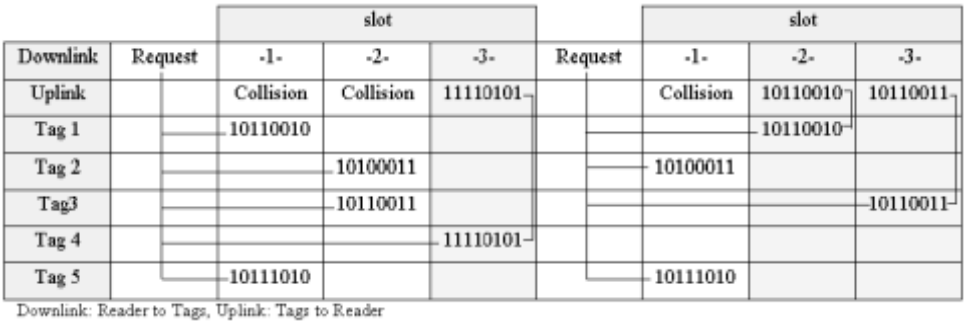
الگوریتم ALOHA ساده ترین نوع الگوریتم ضد تداخل می باشد . ریدر فقط یک دستور درخواست برای تگ ها یی که در ناحیه ی تحت پوشش ریدر قرار دارند ارسال می کند . سپس هر تگ شماره ی ID خود را در یک زمان انتخابی تصادفی ارسال می کند . بنابراین یک احتمال مشخص وجود دارد که در تگ شماره های ID خود را در یک زمان بفرستند و داده آنها با یکدیگر تداخل کند . هر چند اگر فقط یک تگ شماره ID

خود را در زمانی مشخص بفرستد به شناسایی موفق منتهی می شود . در صورتی که تعداد زیادی تگ در میدان ریدر حاضر باشند یا تگ ها حجم زیادی از اطلاعات داشته باشند کارایی این الگوریتم کاهش می یابد . یک امکان برای بهینه کردن کارایی الگوریتم ALOHA , الگوریتم Slotted ALOHA است . در این پروتکل تگ ها فقط در نقاط زمانی همزمان شده ی تعریف شده می توانند شروع به ارسال داده کنند . همزمانی تمام تگ ها باید توسط ریدر کنترل شود .

بنابراین می تواند یک پروتکل ضد تداخل TDMA راه اندازی با ریدر نام گیرد . S ALOHA کارایی بهتری از ALOHA دارد زیرا بین تگ و ریدر همزمانی وجود دارد . علاوه بر اینقاط تعریف شده از زمان که تگ ها پاسخ خود را در آن آغاز می کنند احتمال تداخل را کاهش می دهد . هر چند الگوریتم S ALOHA هنوز این ضعف را دارد که با افزایش تعداد تگ کارایی پایین می آید .

### ۲-۴-۳ الگوریتم BFSA (Basic Frame Slotted ALOHA)

یک فریم یک دروره ی زمانی بین درخواست های یک ریدر است و همچنین شامل تدادی اسلت می باشد . در الگوریتم BFSA در شکل زیر نمایش داده شده است .اندازه فرم در این مثال ۳ اسلات موجود است . در دور اول تگ ۱ و تگ ۵ به صورت همزمان شماره ID خود را در اسلات ۱ می فرستند . تگ های ۲ و ۳ در اسلات ۲ پاسخ می دهند . بدلیل اینکه تگ های ۱ و ۲ و ۳ و ۵ سبب تداخل شده اند , آنها باید در دوره ی خواندن بعدی پاسخ دهند . تگ ۴ در دوره اول می تواند شناسایی می شود زیرا فقط تگ ۴ در اسلات زمانی ۳ پاسخ داده است . در دور دوم , تگ ۱ و تگ ۳ به ترتیب در اسلات ۲ و ۳ شناسایی شده اند هر چند تگ ۲ و ۵ در اسلات ۱ تداخل کردند . بنابراین آنها باید در دور بعدی پاسخ دهند .



شکل : ۲-۲ : نمونه الگوریتم BFSA

گر چه اندازه ی فریم ثابت الگوریتم BFSA موجب پیاده سازی سادهای می شود , عیب آن این است که بازدهی شناسایی تگ کاهش یافته است . برای مثال اگر تعداد زیادی از تگ ها در میدان ریدر قرار داشته باشند , ممکن است در دوره های خواندن هیچ تگی شناسایی نشود و همه ی اسلات ها ممکن است با تداخل پر شوند . همچنین اگر یک اندازه فریم بزرگ استفاده شود , در صورتی که تعداد کمی تگ موجود باشد اتلاف زمانی به وجود می آید .

### ۲-۴-۴ الگوریتم DFSA (Dynamic Frame Slotted ALOHA)

در این الگوریتم اندازه فریم برای شناسایی بهینه تگ ها تغییر می کند . اندازهی فریم می تواند توسط اطلاعاتی از قبیل تعداد اسلات های استفاده شده برای شناسایی تگ , تعداد اسلاتهای تداخل کرده و تعداد اسلات های خالی تعریف شود . پس DFSA می تواند الگگوریتم BFSA را که برای شناسایی تگ ها ناکارآمد بود را بهبود بخشد .

الگوریتم DFSA اندازه فریم را با استفاده از اطلاعات دوره ی خواندن قبلی کنترل می کند , برای مثال تعداد اسلات های خالی , اسلات های تداخل یافته و اسلات های موفق ( اسلات های پر شده با یک تگ ) . اگر احتمال تداخل بیشتر از آستانه ی بالایی باشد , ریدر اندازه ی فریم را افزایش می دهد. در حالی که اگر احتمال کوچکتر از آستانه ی پایینی باشد ریدر فریم را کاهش می دهد . ریدر یک دوره ی خواندن را با حداقل اندازه ی فریم آغاز می کند , بنابراین هنگامی که تعداد تگ ها کم باشد ریدر می تواند تگ ها را به صورت بهینه و بدون افزایش زیاد اندازه ی فریم شناسایی کند . در مورد تعداد زیاد تگ ها ریدر اندازه فریم را به منظور کاهش احتمال تداخل تغییر می دهد .

الگوریتم DFSA از الگوریتم BFSA کاراتر است زیرا ریدر اندازه ی فریم را بر اساس تعداد تگ ها تنظیم می کند . اگر چه تغییر اندازه ی فریم به تنهایی نمی تواند تداخل تگ را در مورد تعداد زیاد تگ ها کاهش دهد زیرا اندازه فریم نمی تواند به طور نامحدود افزایش یابد .

### ۲-۴-۵ الگوریتم ADFSA(Advanced Dynamic Frame Slotted ALOHA)

الگوریتم AFSA اندازه ی فریم را با برآورد کردن تعداد تگ ها تنظیم می کند . پس کارایی بهتری نسبت به الگوریتم BFSA دارد . توابع برآورد زیادی وجود دارند .

آقای Vogt دو روش برای برآورد تعداد تگ ها ارائه کرده ، روش برآورد اول به وسیله ی مشاهده اینکهه یک تداخل حداقل دو تگ مختلف را درگیر می کند بدست آمده . بنابراین یک حد پایین تر (LB) روی مقدار برآورد تعداد تگها می تواند توسط تابع برآورد ساده ای که در معادله ی ۱ نشان داده شده است بدست می آید :

E = ۲ * تعداد اسلات های تداخل یافته (۱)

روش دوم CIILB(Chebyshev Inequality Improved Lower Bound) نامیده می شود و به طریق زیر می تواند بیان شود . نامعادله چبیشف به ما می گوید خروجی یک آزمایش تصادفی یک متغیر تصادفی X در جاهایی نزدیک مقدار مورد انتظار X محتمل تر است . بنابراین یک تابع برآورد متناوب از فواصل بین نتایج خوانده شده که اندازه ی فریم تعداد اسلات های موفق ، تعداد اسلات های تداخل یافنه و تعداد اسلات های بی کار و بردار مقدار مورد نظر برای تعریف تعداد تگ ها استفاده می کند تا فاصله در معادله ۲ حداقل شود .

$$\epsilon_{vd}(N, c_0, c_1, c_k) = min \left| \begin{pmatrix} a_0^{N,n} \\ a_1^{N,n} \\ a_{\geq 2}^{N,n} \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \\ c_k \end{pmatrix} \right|$$

معادله (۲)

که در آن N و n به ترتیب اندازه فریم و تعداد تگ ها را نشان می دهند . a0، a1 و ak به ترتیب تعداد موردانتظار اسلات های خالی ، اسلات های پر شده با یک تگ و اسلات های دارای تداخل می باشند .

روش سوم تابع برآورد توزیع دو جمله ای (BDE) است . مقدار درصد تداخل مورد انتظار در N اسلات عبارتند از :

$$E(C) = 1 - \left(1 - \frac{1}{N}\right)^n \left(1 + \frac{n}{N-1}\right)$$

بعد از پایان یک دوره شناسایی ، درصد واقعی تداخل بدست می آید :

$$C = \frac{collision\ slot\ number}{N}$$

از

$$C = 1 - \left(1 - \frac{1}{N}\right)^n \left(1 + \frac{n}{N-1}\right)$$

می توانیم از رابطه ی بالا تعداد تگ ها ، n را پس از هر دوره ی شناسایی برآورد کنیم .

چهارمین روش از حداکثر شرایط توان عملیاتی برای برآورد تعداد تگ ها ی اطراف استفاده می کند.

اگر اسلاتهای تداخل یافته پس از یک دوره را Coll بنامیم :

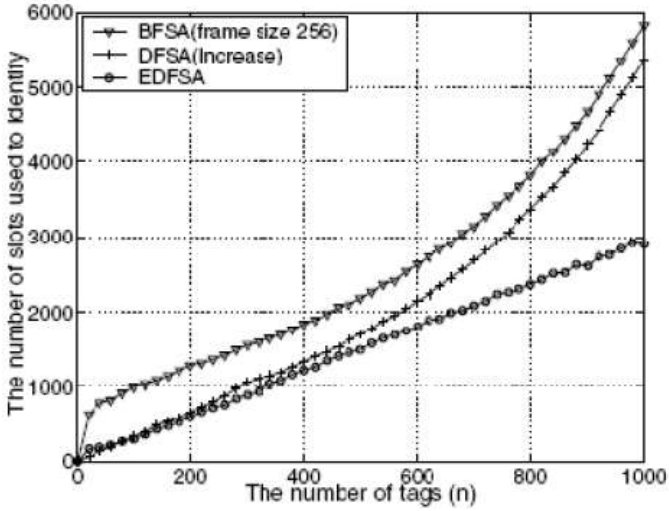
تعداد تگهای برآورد شده = 2.3922 * Mcoll

الگوریتم ADFSA شکل مشابه الگوریتم DFSA دارد که اندازه ی فریم نمی تواند بدون محدودیت افزایش یابد . هنگامی که تعداد تگها زیاد می شود ، بنابراین این الگوریتم هنگامی کارایی خوبی نشان می دهد که تعداد تگ ها کم باشد . در صورتی که تعداد تگ ها زیاد شود کارایی ضعیفی نشان می دهد .

### ۲-۴-۶ گروه بندی تعداد تگ ها ی پاسخ دهنده

در سایر الگوریتم های FSALOHA مشکلی وجود دارد این است که به منظور بازدهی حداکثر سیستم اندازه ی فریم نمی تواند بدون محدودیت زیاد شود . این در حالی است که این مشکل با محدود کردن تعداد تگهای پاسخ دهنده به اندازه ای نزدیک به اندازه ی فریم قابل بررسی است . مثلا الگوریتم EDFSA می تواند این مشکل را گروه بندی تگ های خوانده نشده و با استفاده از رابطه زیر حل کند.

(حداکثر اندازه ی فریم ) N / تعداد تگ های خوانده شده = M تعداد گروه ها



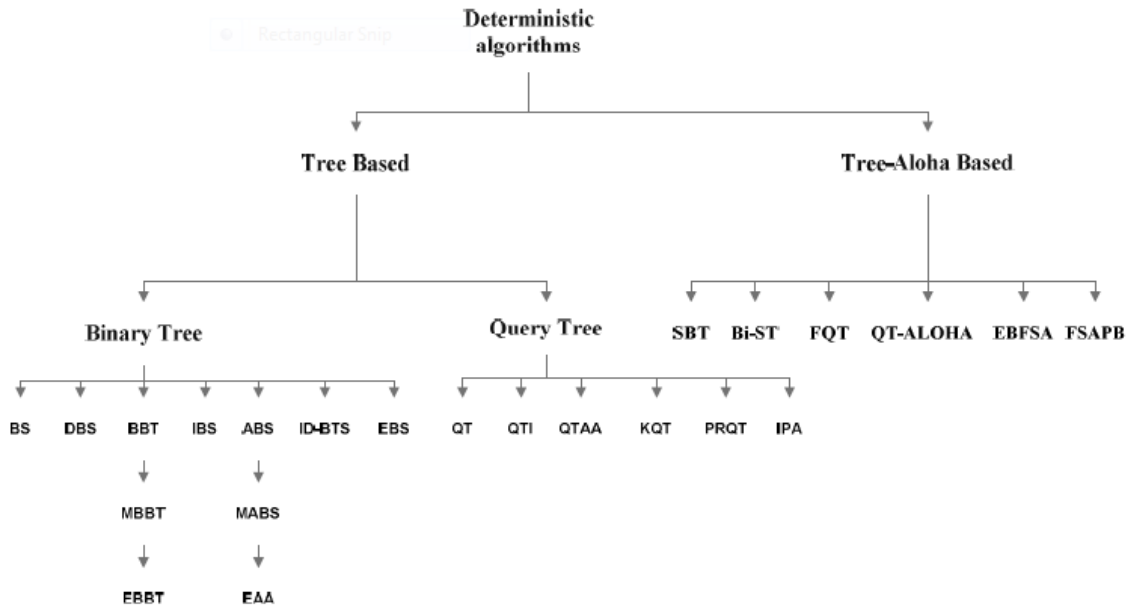

شکل ۲-۳ : مقایسه الگوریتم ها

الگوریتم EDFSA کارایی بیشتر ۱۰۰ و ۸۵ درصدی را به ترتیب نسبت به الگوریتم های BFSA و DFSA نشان می دهد . کارایی الگوریتم VEDFSA با بهبود راه حل های گروه بندی الگوریتم نسبت به الگوریتم EDFSA بهتر است .

گروه در الگوریتم EDFSA ثابت است و در تمام طول فرایند شناسایی تغییر نمی کند ، در حالی که در الگوریتم VEDFSA گروه در تمام طول فرایند شناسایی به طور پویا تغییر خواهد کرد . تگ ها در چندین گروه با هدف کامل شدن کارایی بهینه در هر گروه در دور اول تقسیم شده اند . پس از دور اول خواندن تعدادی از تگ های شناسایی شده از هر گروه کاسته می شود . در دور دوم تگ های خوانده نشده در گروه

های جدید تقسیم می شوند . انتظار می رود هر گروه با بهترین کارایی پر شود . تگ ها در پایان این فرایند شناسایی می شوند .

الگوریتم IFSA راهکار مشابه به الگوریتم EDFSA دارد که تعداد تگهای پاسخ دهنده را برای بدست آوردن کارایی شناسایی عالی محدود می کند ، اما با طرز کاری متفاوت روش IFSA تعداد تگ های پاسخ دهنده را با مقایسه ی یک بخش از شماره شناسایی ID ذخیره شده روی تگ با بیت مقایسه محدود می کند . ریدر بیت مقایسه را برای تگ های حاضر در گستره ی ارتباطی می فرستد . سپس هنگامی که تگ ها این اطلاع را دریافت می کنند بیت مقایسه را با بخشی از ID مقایسه می کنند . بنابراین اگر آن بخش از ID کوچکتر از بیت مقایسه بود ، تگ ها به ریدر پاسخ می دهند .




شکل ۲-۴ : الگوریتم های قطعی

### ۲-۴-۷ الگوریتم Tree Slotted ALOHA

TSA نسخه ی تغییر یافته ی پروتکل S ALOHA می باشد . در این روش تداخل به محض اتفاق افتادن حل می شود , بنابراین اگر تداخل در یک اسلات آشکار شود , ریدر فقط به تگ هایی که در آن اسلات

تداخل کرده اند در دور بعدی درخواست می شوند . دو تگ که در یک فریم تداخل نداشته باشند ممکن است در سایر پروتکل های F S ALOHA در فریم بعدی با هم تداخل کنند .

### ۲-۴-۸ الگوریتم های مبتنی بر درختی

بیشتر الگوریتم های مبتنی بر درخت, اساسا فرم های تغییر یافته ی الگوریتم های Query و باینری هستند .

### ۲-۴-۹ الگوریتم های درختی باینری

در الگوریتم جستجوی باینری تداخل می تواند با کاهش تدریجی بیت های تداخل یافته در شماره سریال تگ ها کاهش یابد . ریدر می تواند موقعیت بیت تداخل یافته را با استفاده از کدینگ دلخواه آشکار کند . برای مثال ,کد منچستر این امکان را می دهد تا تداخل را تا رسیدن به یک بیت مشخص ردیابی کرد . در این رمزنگاری بیت , مقدار بیت می تواند به وسیله ی روش ارسال تعیین شود . اگر ارسال مثبت است به این معنی است که منطق یک است . عدم ارسال به این معنی است که خطایی رخ داده . اگر بیش از یک تگ همزمان بیت هایی با مقادیر مختلف ارسال کنند ارسال مثبت , ارسال منفی بیت های دریافتی را خنثی می کند . بنابراین ارسالی آشکار نمی شود و این به یک خطا منجر می شود . از این رو تداخل در بیت مشخصی می تواند ردیابی شود .

الگوریتم BS تگهای پاسخ دهنده را به دو گروه تقسیم می کند . این الگوریتم به تگ ها اجازه می دهد که بیت تداخل اول ۰ را برای پاسخ داشته باشند . برای دور بعدی , در حالی که به تگ های با اولین بیت تداخل ۱ پاسخ نمی دهند . فرض کنید در ناحیه تحت پوشش ریدر چهار تگ وجود دارد .

در شروع ریدر درخواست می دهد که تگ هایی که شماره سریال هایی کوچکتر یا مساوی 11111111 b دارند پاسخ می دهند . از آنجا که 11111111 b بزرگترین شماره سریال است . همه ی تگ ها با ارسال شماره سریال خود در اولین تکرار به ریدر پاسخ می دهند . تداخلی در شماره سریال دریافتی در بیت ۰ ,

بیت ۴ و بیت ۶ رخ داده است . بنابراین در گستره ی خواندن ۸ تا یا تعداد کمتری تگ قرار دارد . بیت ۶ بالاترین بیت تداخل یافته است و به این معنی است که حداقل یک تگ شمارهسریال کوچکتر از ۱۰۱۱۱۱۱۱ دارد . پس به منظور محدود کردن تعداد تگ های پاسخ دهنده , ریدر در تکرار بعدی درخواست جدیدی برای تگ هایی که شرط ( 10111111 b>) را دارند ارسال می کند . در این مثال شماره سریال دریافتی در بیت ۰ و بیت ۴ تداخل خواهد داشت . از اینرو فرایند تکرار خواهد شد . ریدر به تگ هایی که شماره سریال هایکوچکتر یا مساوی 10101111 bدارند درخواست خواهد داد . فقط تگ ۲ در این تکرار قرار دارد (تکرار سوم ) . سپس ریدر تگ ۲ را انتخاب و ارتباط را آغاز می کند . با تکرار این فرایند ها , همه ی تگ ها می توانند توسط ریدر شناسایی شوند . الگوریتم جداسازی پرس و جوی سازگار Adaptive Query Splitting که مبتنی بر الگوریتم درختی باینری است .

### ۲-۴-۱۰ الگوریتم های Query Tree

در الگوریتم QT ریدر یک پیشوند را مخابره می کند و تگ های اطراف آن پاسخ می دهند . اگر یک تداخل رخ دهد , ریدر درخواست خود را با پیشوندی با یک بیت بلند تر ارسال می کند تا هیچ تداخلی آشکار نشود . دور توسط زیدر خاتمه می یابد و دور جدیدی از درخواست ها با پشوند دیگری آغاز می شود تا یک تگ شناسایی شود . الگوریتم QT پیشرفته (QTI) یک فرم تغییر یافته ی الگوریتم QT است . الگوریتم QTI از درخواست هایی که واقعا به تداخل منجر خواهد شد جلوگیری میکند . به عنوان مثال فرض کنید که یک پرس و جو با پیشوند q یک تداخل تولید کرده و پرس و جو با پیشوند q0 یک اسلات خالی ایجاد کرده است . سپس ریدر از پرس و جوی q1 خودداری می کند و فقط پرس و جوهای q10 و q11 را انجام می دهد .

الگوریتم پیشرفته تهاجمیQT (QTAA) از درخت چهارگانه استفاده می کند نه درخت باینری . برای مثالفرض کنید که پرس و جو با پیشوند q به تداخل منتهی شود ریدر ۴ پرس و جوی q01, q00 , q10 و q11 را ایجاد می کند .

الگوریتم های QT از مزیت بدون حافظه بودن برخور دارند زیرا تگ به حافظه اضافی جز آنچه برای ذخیره ID لازم است , احتیاج ندارد . بنابراین تگ هایی با عملکرد پایین تر و ارزانتر می طلبد . هر چند به دلیل استفاده از پیشوند کارایی این الگوریتم ها به توزیع ID تگ ها محسوس است . الگوریتم پیشوند تصادفی QT(PRQT) با الگوریتم معمولی QT در استفاده از پیشوند های انتخاب شده به صورت تصادفی توسط تگ ها به جای استفاده از پیشوند های مبتنی بر ID همانطور که در QT استفاده می شود متفاوت است . در نتیجه طول و توزیع HD تگ ها دیگر بر روی زمان شناسایی PRQT اثر نمی گذارد .

### ۲-۴-۱۱ الگوریتم های مبتنی بر Tree – ALOHA

کارایی در مسئله ی تداخل تگ ها معمولا به عنوان بازدهی سیستم که اساسا نسبت بین تعداد تگ های مورد شناسایی و تعداد پرس و جو ها یا اسلات های زمانی استفاده شده در سراسر فرایند شناسایی است اندازه گیری می شود . تمام الگوریتم هایی که در بخش های قبلی مرور شد کارایی متوسطی زیر % 50 نشان می دهند . در حقیقت کارا ترین الگوریتم ها یعنی QT و TSA به سور متوسط حدود ۴۰ درصد ظرفیت سیستم را در بهترین کارایی خود بدست می دهند .

کارایی پایین الگوریتم ALOHA به دلیل مشکل Tag Starvation است که پیش از این ارائه شد . زمان شناسایی طولانی در الگوریتم های ضد تداخل بر اساس درخت ویژگی ذاتی روش است و مخصوصا با تعداد زیاد تگهای با طول ID زیاد ظاهر می شود . از طرف دیگر الگوریتم های ضد تداخل تگ RFID بر اساس درختی , نرخ کامل خواندن را بدست می دهند و الگوریتم های ضد تداخل بر اساس ALOHA پیاده سازی ساده و کارایی خوبی را برای تعداد کمی از تگ ها بدست می دهد . بنابراین ترکیب هر دو روش و بهره مندی از مزایای آنها می تواند به کارایی بهتر منجر شود .

### ۲-۴-۱۲ الگوریتم Slotted Binary Tree

این الگوریتم تداخل را در همان زمان بروز رفع می کند , هنگامی که یک تداخل در اسلات i رخ دهد . تمام تگ ها که در اسلات تداخل یافته ی i پاسخ نداده اند منتظر می مانند تا تداخل بررسی شود , سپس ریدر از

تگ هایی که در تداخل گرفتار شده اند درخواست می کند که به صورت اتفاقی گروه صفر یا گروه یک را انتخاب کنند . تگ هایی که گروه صفر را انتخاب کرده اند در اسلات i+1 پاسخ می دهند در حالی که تگ های گروه دیگر منتظر می مانند تا تمام تگ های موجود در گروه صفر با موفقیت شناسایی شوند . اگر اسلات i+1 بی کار باشد یا اسلات موفق باشد , تگ های گروه ۱ در اسلات i+2 پاسخ می دهند . در غیر این صورت تداخل وجود دارد ؛ برای رفع تداخل جدید روند مشابهی تکرار می شود .

اگر n تگ در منطقه ی تحت پوشش ریدر وجود داشته باشد , تکرار الگوریتم SBT (Isbt) توسط فرمول زیر می تواند محاسبه شود :

$$I_{SBT} = 1 + \sum_{k=2}^{n} \binom{n}{k} \frac{2(k-1)(-1)^k}{[1-p^k-(1-p)^k]}$$

### ۲-۴-۱۳ الگوریتم ضد تداخل مبتنی بر Bi-slotted tree

الگوریتم‌های مبتنی بر Bi-slotted tree که در این بخش توضیح داده خواهند شد ، الگوریتم‌های درخت پرس و جوی دو اسلاتی (BSQTA) و الگوریتم درخت ردیابی تداخل دو اسلاتی (BSCTTA) هستند.

جنبه‌ی اصلی این الگوریتم‌ها این است که آن‌ها بیتهایی در دو دسته‌ی با طول n ارسال می‌کنند که n-1 بیت مشابه دارند و بیت آخر متفاوت است. گام‌های زیر فرآیند شناسایی این دو الگوریتم را نشان می‌دهند :

۱- درخواست: ریدر پیشوندی با طول n-1 مخابره می‌کند .

۲- گروه بندی: تگ‌های موجود در ناحیه‌ی تحت پوشش ریدر اگر پیشوند با n-1 بیت اول ID آن‌ها مطابقت داشته باشد پاسخ می‌دهند. این تگ‌ها بسته به بیت nام در یکی از دو اسلات زمانی پاسخ می‌دهند. اگر بیت nام ۰ باشد تگ‌ها اسلات اول و اگر ۱ باشد اسلات دوم را انتخاب میکنند. بنابراین اسلات انتخاب شده مقدار بیت nام را نشان می‌دهد.

- BSQTA : تگ ما IDهای خود را از بیت n+1ام تا بیت پایانی ارسال می‌کنند.

- BSCTTA : تگ ما IDهای خود را از بیت n+1ام تا زمانی که آن‌ها علامت ACK را دریافت کنند ارسال می‌کنند ، که این علامت می‌گوید تداخلی از ریدر وجود ندارد.

۳- تصمیم گیری:

- اگر تداخل رخ داد ، ریدر یک پیشوند جدید در (LIFO) ذخیره می‌کند.

BSQTA.i : اتصال پیشوند n-1 بیتی و تشریح اسلات انتخاب شده و بیت‌های دریافت شده قبل از رخ دادن تداخل.

- اگر آخرین بیت تداخل یافته است ، ریدر در نظر می‌گیرد که دو تگ وجود دارد.
- اگر تداخلی وجود ندارد ، تگ توسط ریدر شناسایی می‌شود.

۴- گام‌ها تا زمانی‌که LIFO خالی شود تکرار می‌شود.

با استفاده از BSQTA و BSCTTA در مجموع متوسط پیشوندهای مورد نیاز نسبت به الگوریتم‌های مبتنی بر درخت QTA و CTTA به نیم سر جمع پیشوندها کاهش می‌یابد. مشکل کارایی بهتر BSQTA و BSCTTA از سایر الگوریتم‌های ضد تداخل مبتنی بر درختی را نشان می‌دهد.

در مقایسه بین BSQTA و QTA متوسط پیشوند مورد نیاز در BSQTAتقریبا به ۵۰٪ پیشوندهای مورد نیاز در QTA کاهش یافته ، در حالی که تغییری در سطح متوسط بیت‌های پاسخ مورد نیاز برای شناسایی یک تگ وجود ندارد. بنابراین بیت‌های کمتری در BSQTA برای شناسایی یک تگ نسبت به QTA نیاز است. علاوه بر این تفاوت کارایی بین BSQTA و QTA در تعداد بیت‌های مورد نیاز برای شناسایی یک تگ ، هنگامی که تعداد تگ‌ها افزوده می‌شود افزایش می‌یابد. همچنین به منظور ارزیابی کارایی سیستم با استفاده از معیاری دیگر ، تعداد تکرر برای شناسایی یک تگ اندازه‌گیری شده است.

BSQTAبهتر از QTA در هر دو مورد متوسط بیت‌های مورد نیاز و متوسط تکررهای مورد نیاز برای شناسایی یک تگ عمل می‌کند.

الگوی مشابهی در مقایسه‌ی BSCTTA و CTTA وجود دارد. BSCTTA نسبت به CTTA در بیت‌های مورد نیاز برای شناسایی یک تگ کارایی بهتری بدست می‌آید.

متوسط پیشوندهای مورد نیاز در BSCTTA حدودا به نیمی از پیشوندهای مورد نیاز در CTTAکاهش یافته ، در حالی که تغییری در متوسط بیت‌های پاسخ مورد نیاز برای شناسایی یک تگ بوجود نیامده بنابراین در BSCTTA نسبت به CTTA بیت‌های کنترلی برای شناسایی یک تگ نیاز است. علاوه بر این BSCTTA به نیمی از تعداد تکررهای CTTA نیاز دارد.

روشن است که BSQTA و BSCTTA در فرآیند شناسایی به ترتیب از QTA و CTTA سریعتر عمل می‌کنند.

### ۲-۴-۱۴ الگوریتم درخت پرس و جوی قاب بندی شده framed query true (FQT)

در این الگوریتم تگ‌ها بصورت اتفاقی به دو دسته‌ی فریم تقسیم بندی می‌شوند. الگوریتم QT برای شناسایی تگ‌های درون هر دسته استفاده می‌شود. فرآیند الگوریتم FQT در زیر شرح داده شده است.

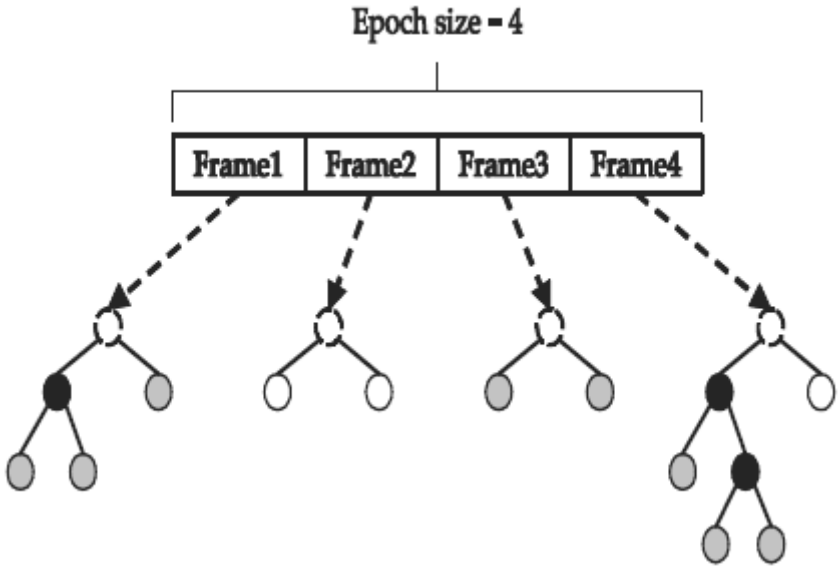
در آغاز فرآیند شناسایی ، ریدر یک درخواست برای ID به تمام تگ‌های در منطقه‌ی تحت پوشش ریدر ارسال می‌کند. این درخواست شامل اندازه‌ی یک دوره است که تعداد کل فریم‌ها است (شکل ۱۳ را ببینید). سپس هر تگ به صورت اتفاقی یک فریم را انتخاب می‌کند و فقط هنگامی که ریدر فریم خود را مورد پرس و جو قرار می‌دهد پاسخ می‌دهد. در درون هر فریم ریدر از الگوریتم QT برای شناسایی تگ‌های این فریم استفاده می‌کند. بنابراین اندازه‌ی دوره را مانند پیشوند ID به تگ‌ها می‌فرستد. اگر فریم انتخابی مشابه آنچه فرستاده شده باشد ، تگ ID با پیشوند مطابقت داده شده خود را پاسخ می‌دهد (الگوریتم QT). هنگامی که تمام تگ‌های داخل فریم شناسایی شدند ، ریدر با فریم بعدی ادامه می‌دهد. فرآیند بالا برای تمام فریم‌ها تکرار می‌شود. شکل ۱۳ یک مثال از فرآیند شناسایی الگوریتم FQT را هنگامی که تعداد تگ‌های در اطراف ۸ و تعداد فریم‌ها ۴ باشد ، نشان می‌دهد.

برای افزایش کارایی الگوریتم FQT یک اندازه‌ی دوره‌ی مناسب باید تعیین شود.

الگوریتم QT بهترین کارایی را زمانی دارد که عمق درخت از ۲ تجاوز نکند. (فریم ۳ در شکل ۱۳)

هنگامی که تعداد تگ‌های مورد شناسایی N است و اندازه‌ی دوره ES است. ES ایده‌آل میتواند از رابطه‌ی زیر تعیین شود.

$$N=2*ES \quad (9)$$




شکل ۲-۵ : نمونه الگوریتم FQT

این در حالی است که تعیین اندازه‌ی دوره‌ای مناسب ، در آغاز فرآیند شناسایی بدلیل نامعین بودن N دشوار است. بنابراین الگوریتم نهایی FQT از FFT (first frame test) استفاده می‌کند. FFT فرآیند را از تعداد کمی از فریم‌ها آغاز می‌کند و اگر تداخل در فریم اول از یک آستانه‌ی تداخل تجاوز کند ، FFT اندازه‌ی دوره را افزایش می‌دهد همانطور که اشاره شد تمام تگ‌ها بصورت اتفاقی در فریم‌ها تقسیم می‌شوند. اگر تعداد زیادی تداخل در فریم اول وجود داشته باشد ، به احتمال زیاد سایز فریم‌ها هم هم شرایط مشابهی دارند.

فریم ۳ در شکل ۱۳ بهترین کارایی الگوریتم را هنگامی که تعداد تگ‌ها ۲تا و با عمق ۱ است نشان می‌دهد. بنابراین به منظور محافظت درخت از عمیق‌تر شدن بیش از این آستانه‌ای لازم است.

## ۲-۴-۱۵ الگوریتم (QT-ALOHA)

الگوریتم QT-ALOHA ترکیبی از الگوریتم FS-ALOHA و الگوریتم QT است.

الگوریتم FQT اساسا الگوریتم FS-ALOHA را می‌سازد و در حقیقت از الگوریتم QTبعنوان فرآیند اصلی شناسایی استفاده می‌کند. با این حال الگوریتم QT-ALOHA از الگوریتم FS-ALOHA به عنوان فرآیند شناسایی تگ حقیقی استفاده می‌کند و الگوریتم QT به عنوان یک تصویر بزرگ فرآیند الگوریتم QT-ALOHA در ادامه توضیح داده می‌شود.

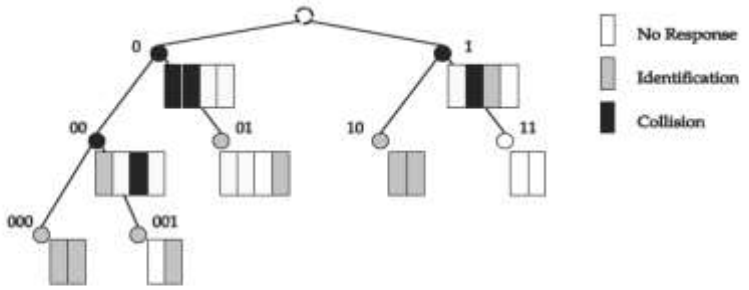
در آغاز فرآیند ، ریدر درخواستی شامل یک پیشوند و یک اندازه‌ی فریم را با یکدیگر می‌فرستد. سپس تنها تگ‌های با ID قابل تطبیق با الگوریتم FS-ALOHA عمل می‌کنند و در اندازه‌ی فریم فرستاده شده پاسخ می‌دهند. در حین اجرای الگوریتم FS-ALOHA اگر تداخلی حتی در یک اسلات رخ دهد ، بهنوان یک تداخل الگوریتم QT در نظر گرفته می‌شود. بنابراین یک پیشوند جدید تولید شده و به صف اضافه می‌شود.

مثال در شکل ۱۴ فرآیند الگوریتم QT-ALOHA را نشان می‌دهد. در این مثال فرض شده تعداد تگ‌هایی که باید شناسایی شوند ۸ و اندازه‌ی فریم اولیه از ۴ شروع می‌شود. جدول در شکل ۱۴ باقی فرآیند را نشان می‌دهد.

شکل ۱۵ و۱۶ نشان می‌دهد که الگوریتم FQT به صورت خاص افزایش کارایی زیادی را به تصویر می‌کشد.

شکل ۱۵ نشان می‌دهد که چند پرس و جو و پاسخ برای شناسایی موفق یک تگ نیاز است. در مقایسه‌ی بین الگوریتم‌ها ، FQT بهترین کارایی را نسبت به سایر الگوریتم‌ها را نشان می‌دهد که در حدود ۱۰ تا ۵۰ درصد از زمان بسیاری دیگر از الگوریتم‌های ضد تداخل موجود است.

شکل : ۲-۶ :




| Query | 0 | 1 | 00 | 01 | 10 | 11 | 000 | 001 |
|---|---|---|---|---|---|---|---|---|
| Frame Size | 4 | 4 | 4 | 4 | 2 | 2 | 2 | 2 |
| Response | collision | collision | collision | identified | identified | no res. | identified | identified |

نمونه QT ALOHA

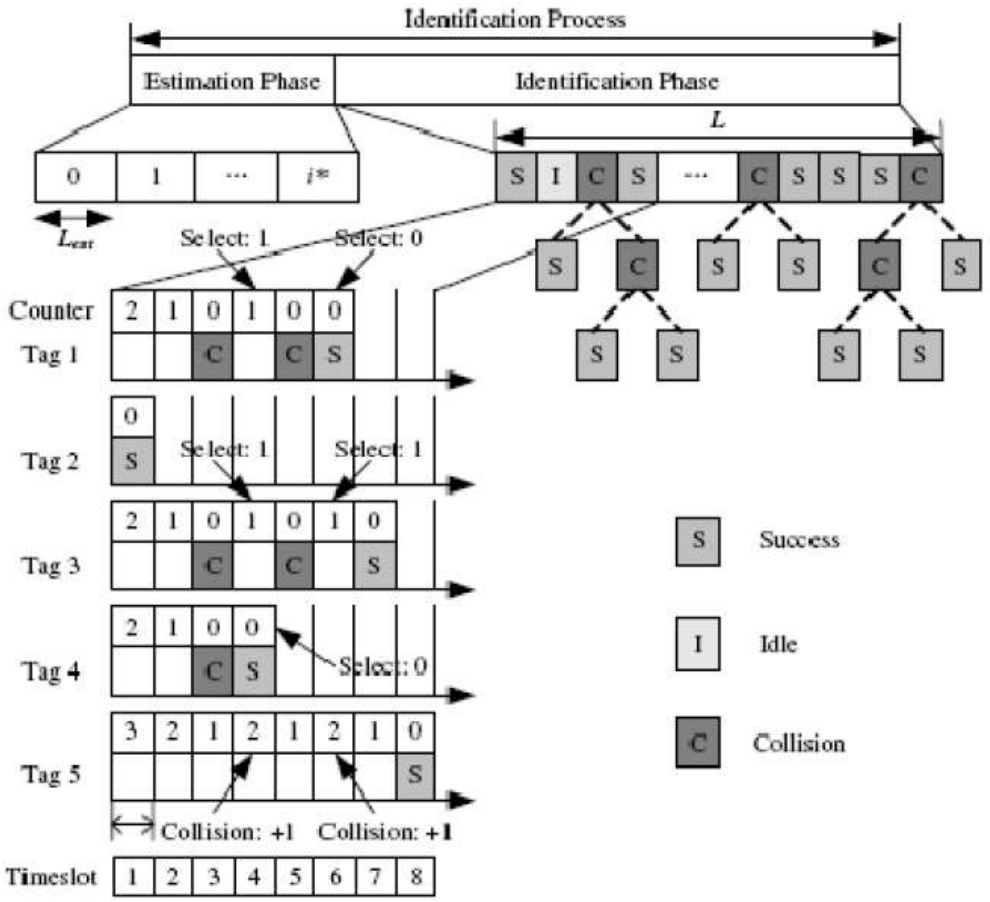
## ۲-۴-۱۶ الگوریتم FS-ALOHA با برآورد تگ و جداسازی (EBFSA)

EBFSA دو فاز دارد: فاز برآورد و فاز شناسایی. ریدر تعداد تگ‌های در همسایگی در فاز برآورد را برآورد می‌کند و تگ‌ها IDهای خود را در خلال فاصله‌ی زمانی فریم در فاز شناسایی می‌فرستند. فاز برآورد ، برآورد تگ را با (DFSA) هدایت می‌کند و از اندازه‌ی فریم ثابت $L_{est}$ استفاده می‌کند. (شکل ۱۷ را ببینید).

اگر احتمال تداخل بیش از آستانه‌ی $P_{coll-th}$ باشد ، اندازه‌ی فریم توسط ماسک بیت در یک فریم پرس‌وجو با ضریبی از $F_d$ کاهش می‌یابد. این فرآیند تا هنگامی که احتمال تداخل $P_{coll}$ از $P_{coll-th}$ کمتر شود تکرار می‌شود. سپس تعداد تگ‌ها ، n ، از $P_{coll}$ و $L_c = L_{est}$ یا استفاده از معادله‌ی ۱۰ برآورده می‌شود.

$$P_{coll} = 1 - P_{idle} - P_{succ} = 1 - \left(1 - \frac{1}{LC}\right)^n - n\frac{1}{LC}(1 - \frac{1}{LC})^{n-1}$$

در این الگوریتم روش برآورد تگ می‌تواند صرفنظر از اندازه‌ی فریم اولیه و تعداد واقعی تگ‌های مورد شناسایی محاسبه می‌شود. پس از آن فاز شناسایی آغاز می‌شود. هر تگ به طور اتفاقی یک مقدار شمارنده را انتخاب می‌کند. سپس ریدر ، اندازه‌ی فریم بهینه (L) را بسته به تعداد تگ‌های برآورده شده در فاز برآورد تعیین می‌کند.




شکل ۲-۷ : نمونه ی الگوریتم EBFSA

پس از آن‌که هر تگ بصورت اتفاقی یک تعداد شمارنده را انتخاب کرده در هر اسلات زمانی تگ مقدار شمارنده‌ی متفاوتی دارد. تگ برای هر اسلات زمانی مقدار شمارنده‌ی خود را یکی کاهش می‌دهد. تگ ID خود را تا زمانی‌که شمارتده‌ی آن بار دیگر صفر شود ارسال می‌کند. اگر تداخلی موجود باشد ، الگوریتم با انتخاب باینری با تداخل رفتار می‌کند. بنابراین تگ‌های تداخل یافته مقدار شمارنده‌های خود را بصورت اتفاقی از میان ۰ و ۱ انتخاب می‌کنند.

این در حالی است که سایر تگ‌ها شمارنده‌ی خود را یک واحد افزایش می‌دهند ، بعنوان مثال شکل ۱۷ نشان می‌دهد که تگ‌های ۱ و۲ و۴ در اسلات زمانی ۳ تداخل کرده‌اند و برای مقابله با این شکل ، آن‌ها ۰ یا ۱ را بطور اتفاقی انتخاب می‌کنند و تگ ۵ شمارنده‌ی خود را ۱ واحد اضافه می‌کند. در اسلات زمانی ۴ ، تگ

۴ با موفقیت شناسایی شده ، چرا که شمارنده‌ی آن صفر است. فرآیندهای مشابهی ادامه می‌یابد تا همه‌ی تگ‌ها با موفقیت شناسایی شوند .

در الگوریتم DFSA بسیاری از فریم‌ها به هدر می‌روند ، زیرا تگ‌های تداخل یافته IDهای خود را به منظور حل مشکل تداخل در چندین فریم می‌فرستند. این در حالی است که در الگوریتم EB-FSA ، اندازه‌ی فریم بطور صحیح در فاز برآورد تعیین شده است و توسط جداسازی باینری بدون ایجاد فریم‌های بیشتر در فاز شناسایی با تداخل رفتار شده است. بنابراین فرآیند شناسایی در الگوریتم EB-FSA سریع است.

همانطور که در شکل ۱۸ نشان داده شده ، EB-FSA از ۲۴۳۳ اسلات استفاده میکند که ۱۰.۲٪ کوچکتر از DFSA (۲۷۰۹ اسلات) و ۱۵.۲٪ کوچکتر از الگوریتم درخت باینری (۲۸۶۹ اسلات) است.

### ۲-۴-۱۷ الگوریتم FSA با فریم راهنما و انتخاب باینری (FSAPB)

در الگوریتم FSAPB ، از معادله‌ی ۱۰ برای برآورد تعداد تگ ها استفاده می‌شود.

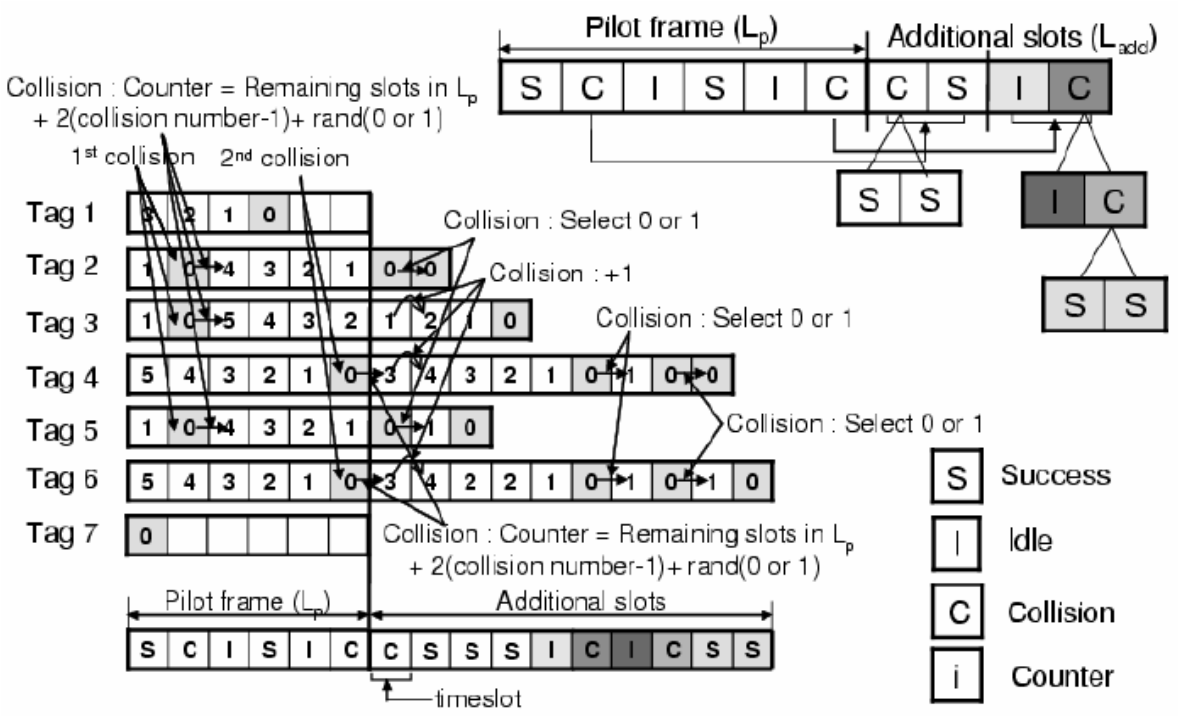
در آغاز تگ‌های پاسخ دهنده به وسیله‌ی ماسک‌های بیت به منظور کاهش تعداد اسلات‌های زمانی به Mزیرگروه تقسیم می‌شوند. سپس فریم راهنما (LP) فقط با زیر گروه اول استفاده می‌شود. سایر زیرگروه‌ها تعداد تگ‌هایی که از LP استفاده نمی‌کنند را تعیین می‌کنند. ریدر یک اسلات زمانی اضافی در انتهای LP برای رفع تداخل‌های رخ داده در LP ایجاد می‌کند. الگوریتم FSAPB می‌تواند در درون دسته‌هایی با احتمال تداخل کم و زیاد گروه تبدیل می‌شود. تمام تگ‌ها IDهای خود را در LP می‌فرستند. سپس هنگامی که LP پر شده است ، $P_{coll}$ محاسبه شده و با آستانه‌ی $P_{th}$ مقایسه می‌شود.

در شرایط احتمال کم تداخل ، $P_{coll}$ کو چکتر از $P_{th}$ است ، الگوریتم درخت باینری و اسلات زمانی اضافه‌ی $L_{add}$ استفاده می‌شود.

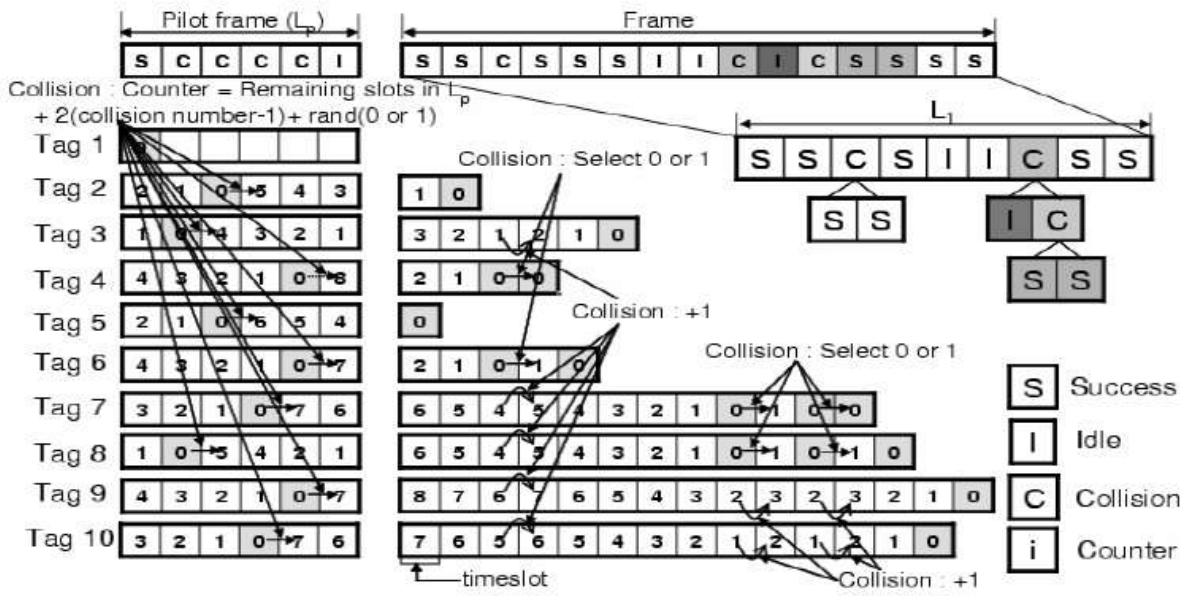
شکل ۱۹ یک مثال از حالت تداخل کم را نشان می‌دهد. تگ‌ها هنگامی‌که شمارنده‌هایشان صفر شود ، IDهای خود را می‌فرستند ، تگ ۱ و تگ ۷ با موفقیت در LP شناسایی شده‌اند .تگ‌های ۲ و ۳ و ۵ در LP

تداخل کرده‌اند. بنابراین دو اسلات زمانی اضافه به LP اضافه شده است و آن تگ‌ها در طول $L_{add}$ با الگوریتم درخت باینری عمل می‌کنند.

تگ ۲ و ۵ دوباره در $L_{add}$ تداخل کرده‌اند. آن‌ها از الگوریتم درخت باینری استفاده می‌کنند و سایر تگ‌ها شمارنده‌ی خود را ۱ واحد افزایش می‌دهند.




شکل ۲-۸ : نمونه الگوریتم FSAPB ، وقتی Pcoll کمتر از Pth باشد




شکل ۲-۹ : نمونه الگوریتم FSAPB ، وقتی Pcoll بیشتر از Pth باشد

در احتمال تداخل زیاد هنگامی‌که $P_{coll}$ بزرگتر از $P_{th}$ است. بیشتر اسلات‌های زمانی $L_p$ ،احتمال بیشتری دارد که تداخل داشته باشند زیرا تعداد تگ‌ها زیاد است.

پس برای شناسایی تگ یک فریم جدید $L_1$ استفاده شده است. اندازه‌ی $L_1$می‌تواند با حداقل n که با معادله‌ی ۱۰ هماهنگی دارد تنظیم شود.

شکل ۲۰ یک مثال از احتمال تداخل زیاد در الگوریتم FSAPB را نشان می‌دهد ، تگ‌های تداخل یافته در $L_p$ID های خود را در $L_1$ دوباره می‌فرستد. تگ‌های ۴ و ۶ در اسلات زمانی ۳ تداخل کرده‌اند و سایر تگ‌ها در $L_1$ شمارنده‌ی خود را یک واحد افزایش می‌دهند. این فرآیند برای سایر اسلات‌های زمانی تکرار می‌شود.

تعداد تگ‌ها پس از اینکه تگ‌های موجود در زیر گروه اول شناسایی شدند می‌توانند تعیین شوند ، زیرا ماسک بیت تگ‌ها را بصورت یکنواخت جدا می‌کند. اندازه‌ی فریم مناسب زیرگروه جاری برابر است با تعداد تگ‌های شناسایی شده در زیر گروه قبلی ضربدر یک ثابت (r)تا هنگامی که یک اندازه‌ی فریم مناسب تعیین شود. الگوریتم درخت باینری اعمال می‌شود.

در الگوریتم DFSA اندازه‌ی فریم بهینه با تعداد تگ‌ها برابر است. درحالی که اندازه‌ی فریم بهینه در الگوریتم FSAPB با تعداد تگ‌ها برابر نیست (شکل ۲۱ را ببینید). بنابراین معادله‌ی زیر برای محاسبه‌ی اندازه‌ی فریم بهینه‌ی جدید $L_{opt}$ مورد استفاده قرار می‌گیرد.

$$T = \sum_{k=0}^{n} nC_k \left(\frac{1}{L_{opt}}\right)^k \left(1-\frac{1}{L_{opt}}\right)^{n-k} L_{opt}(\alpha_k + 1) \qquad (11)$$

که در آن T ، n و k به ترتیب ، تعداد کل اسلات‌های زمانی ، تعداد تگ‌ها و تعداد تگ‌های ارسال شده در یک اسلات زمانی است. علاوه بر این $\alpha_k$ متوسط تعداد اسلات‌های زمانی استفاده شده توسط درخت باینری را هنگامی که تداخل رخ می‌دهد نشان می‌دهد. اندازه‌ی فریم بهینه برای nهای مختلف با مشتق گیری از معادله‌ی ۱۱ تعیین می‌شود.

$L_{opt}$ از شکل ۲۲ می‌تواند 0.88n تعیین شود.

شکل ۲۳ تعداد اسلات‌های زمانی هدر رفته برای برآورد تگ را برای تعداد مختلف تگ‌ها نشان می‌دهد. می‌توان دید که FSAPBاز اسلات‌های زمانی کمتری نسبت به EBFSA برای برآورد تگ‌ها نشان می‌کند.

هنگامی که تعداد تگ‌ها ، n ، کمتر از ۲۰۰ است ، $P_{coll}$ در $L_p$کمتر از آستانه است. بنابراین ریدر از $L_{add}$ استفاده می‌کند ، که به دلیل تعداد کم تگ‌ها در $L_p$ به خوبی توسط الگوریتم باینری انجام می‌شود. در حالی که تعداد تگ‌ها از ۳۰۰ به ۱۰۰۰ عدد تغییر می‌کند ، به خاطر تعداد تداخل زیاد در $L_p$ از $L_1$ به جای $L_{add}$ برای شناسایی تگ استفاده می‌شود. بنابراین FSAPB اسلات زمانی بیشتری نسبت به $L_p$ استفاده نمی‌کند. در حالی که EBFSA هنگامی که تعداد تگ‌ها زیاد است به تعداد زیادی اسلات زمانی احتیاج دارد ، هنگامی که n=1000 ، EBFSA ۵.۵ برابر FSAPB اسلات زمانی احتیاج دارد.

با مقایسه‌ی FSAPB با الگوریتم مبتنی بر ALOHA و درخت باینری شکل ۲۴ تعداد اسلات‌های زمانی استفاده شده برای شناسایی تگ‌ها را به ازای تعداد متفاوت تگ‌ها نشان می‌دهد. به ازای تعداد تگ‌ها ،بیشتر

الگوریتم‌ها تقریبا کارایی مشابهی دارند. در حالی که با افزایش تعداد تگ‌ها FSAPB بهترین کارایی را نسبت به سایر الگوریتم‌ها بدست می‌آید. با در نظر گرفتن ۱۰۰۰ تگ ، EBFSA ، DFSA و درخت باینری ، اسلات‌های بیشتری در حدود ۷.۳٪ ، ۱۴.۲٪ و ۲۱.۹٪ به ترتیب نسبت به FSAPB استفاده می‌کنند.

# فصل سوم

# بررسی استاندارد ISO/IEC 14443

**مقدمه**

در این فصل به توضیح و تفسیر استاندارد ISO/IEC 14443 نوع A می پردازیم . این استاندارد به چهار بخش تقسیم می شود که در هر بخش به شرح قسمتی از این استاندارد می پردازیم . در بخش اول به تفسیر ویژگی های فیزیکی کارت ها و ریدر می پردازیم . در بخش دوم توان فرکانس رادیویی و سیگنال ارتباطیبین ریدر و کارت را مورد بررسی قرار می دهیم . در بخش سوم به برسی راه اندازی این پروتکل و همچنین به قسمت ضد تداخل می پردازیم . در بخش آخر نیز به فعال سازی و غیر فعال سازی این پروتکل PICC نوع A می پردازیم .

# استاندارد 14443-1

## کارت های شناسایی – مدارات مجتمع کارت های بدون تماس – کارت های نزدیک

## قسمت اول : ویژگی های فیزیکی

معرفی :

ISO/IEC 14443 یکی از استاندارد های بین المللی هست که به تتفسیر پارامترهای کارت های شناسایی و استفاده آنها می پردازد .

در این قسمت از ISO/IEC 14443 ویژگی های فیزیکی توضیح داده خواهد شد .

این استاندارد بین المللی مانع همکاری دیگر تکنولوژی های استاندارد بر روی کارت نمی شود .

استاندارد های کارت های بدون تماس انواع گسترده ای از استاندارد ها مانند ISO/IEC 10536 , ISO/IEC 14443 و ISO/IEC 15693 را پوشش می دهد . این استاندارد ها برای عملیات های با فاصله ی خیلی کم , کم و زیاد تعبیه شده اند .

## ۱ هدف

در این قسمت از ISO/IEC 14443 ویژگی های فیزیکی کارت های نزدیک مشخص می شود . این از کارت های شناسایی نوع ID-1 می خواهد که به دستگاه کوپلینگ نزدیک شود .

این قسمت از ISO/IEC 14443 بایستی در همزمان با دیگر قسمت های ISO/IEC 14443 استفاده شود .

**۲** استاندارد های زیر شامل قوانینی می شوند که در این متن ، قوانین این قسمت از استاندارد ISO/IEC 14443 را تشکیل می دهد .

ISO/IEC 14443 کارت های شناسایی – کارتهای بدون تماس مدار مجتمع – کارتهای نزدیک

ISO/IEC 10373 کارت های شناسایی – روش های آزمون

ISO/IEC 7816-2 کارت های شناسایی – کارت های مجتمع شده با تماس

**۳ تعاریف , اختصارها و نماد ها**

**۳-۱ تعاریف**

در ادامه این کلمات به این معانی هستند :

**۳-۱-۱ Integrated circuit (IC)**

قطعات الکترونیکی که برای انجام پردازش و توابع حافظه طراحی شده اند .

**۳-۱-۲ بدون تماس (Contactless)**

مربوط به انجام تبادل سیگنال با کارت بدون استفاده از المان های گالوانیکی (یعنی عدم حضور مسیر مستقیم از تجهیزات رابط خارجی به مدارات مجتمع درون کارت ) .

**۳-۱-۳ کارت های مجتمع شده بدون تماس (Contactless integrated circuit card)**

یک کارت نوع ID-1 که درون آن مدار مجتمع قرار گرفته شده که کار برقراری تماس بدون تماس را انجام می دهد .

**۳-۱-۴ PICC**

یک کارت نوع ID-1 که مجتمع شده و وسایل ارتباط بر روی آن قرار داده شده است و با دستگاه کوپلینگ توانایی ارتباط دارد .

**۳-۱-۵ PCD**

دستگاهی که می خواند/ می نویسد و همچنین از مدار کوپلینگ سلفی برای تامین توان کارت ها استفاده می کند .

## ۴ ویژگی های فیزیکی

### ۴-۱ کلیت

PICC باید ویژگی های فیزیکی مورد نیاز مشخص شده برای ID-1 در ISO/IEC 7810 را داشته باشد .

### ۴-۲ ابعاد

ابعاد باید مطابق ابعاد مشخص شده در ISO/IEC 7810 برای کارت های نوع ID-1 باشد .

### ۴-۳ ویژگی های اضافه شده :

#### ۴-۳-۱ نور ماورا بنفش

PICC ها باید در مقابل اثرات نور های ماورا بنفش که سطح آنها از یک روز معمولی در سطح دریا بالاتر است محافظت شوند. جایی که محافظت بیشتری مورد نیاز باشد این مسئولیت کارخانه سازنده کارت است تا آن را مهیا سازد .

#### ۴-۳-۲ امواج X

PICC باید بعد از قرار گرفتن در معرض موج X با انرژی 100 keV و دز انباشته 0.1 Gy در سال به کار خود ادامه دهد .

#### ۴-۳-۳ قابلیت انعطاف پذیری (خمیدگی)

PICC باید بعد از آزمون های در استاندارد ISO/IEC 10373 به طور معمول به کار خود ادامه دهد .

#### ۴-۳-۴ قابلیت انعطاف پذیری (پیچشی)

PICC باید بعد ار قرار گرفتن در آزمون توضیح داده شده در ISO/IEC 10373 جایی که زاویه چرخش برابر ۱۵ درجه است به طور معمول به کار خود ادامه دهد .

#### ۴-۳-۵ میدان های مغناطیسی متناوب

الف ) PICC باید بعد از قرار گیری در معرض یک میدان مغناطیسی با سطح متوسط داده شده در جدول زیر به کار خود ادامه دهد :

| Frequency Range (MHz) | Average Magnetic Field Strength (A/m) | Averaging Time (minutes) |
|---|---|---|
| 0,3 - 3,0 | 1,63 | 6 |
| 3,0 - 30 | 4,98/f | 6 |
| 30 - 300 | 0,163 | 6 |

ماکزیمم مقدار میدان مغناطیسی به ۳۰ برابر مقدار متوسط محدود است .

ب ) PICC باید بعد از قرار گیری در معرض یک میدان مغناطیسی 12 A/M در فرکانس ۱۳.۵۶ مگاهرتز به کار خود ادامه دهد .

### ۴-۳-۶ میدان الکتریکی متناوب

PICC باید بعد از قرار گیری در معرض یک میدان الکتریکی با مقدار متوسط داده شده در جدول زیر به کار خود ادامه دهد :

| Frequency Range (MHz) | Average Electric Field Strength (V/m) | Averaging Time (minutes) |
|---|---|---|
| 0,3 - 3,0 | 0,614 | 6 |
| 3,0 - 30 | 1842/f | 6 |
| 30 - 300 | 61,4 | 6 |

ماکزیمم مقدار میدان الکتریکی به ۳۰ برابر مقدار متوسط محدود است .

### ۴-۳-۷ الیکتریسیته ساکن

PICC باید بعد از قرار گرفتن در میدان ۶ کیلوولتی به کار خود به طور معمول ادامه دهد .

### ۴-۳-۸ میدان مغناطیسی ساکن

PICC باید بعد از قرار گرفتن در میدان مغناطیسی ساکن 640 kA/m به کار خود ادامه دهد .

### ۴-۳-۹ دمای کار

PICC باید در محدوده دمای محیط ۰ تا ۵۰ درجه سانتی گراد به طور صحیح کار کند .

# استاندارد 14443-2

## کارت های شناسایی – مدارات مجتمع کارت های بدون تماس – کارت های نزدیک

## قسمت دوم : توان فرکانس رادیویی و سیگنال رابط

### ۱ – هدف

این قسمت از ISO/IEC 14443 طبیعت و ویژگی های ذاتی میدان هایی که برای تهیه قدرت و ارتباط دو طرفه بین دستگاه ریدر و کارت استفاده می شوند را مشخص می کند .

این قسمت از ISO/IEC 14443 باید با ترکیب دیگر قسمت های این استاندارد استفاده شود .

در این قسمت از ISO/IEC 14443 ملزومات تولید میدان های کوپلینگ و همچنین ملاحظات تشعشعات الکترومغناطیسی و خطرات آن بر روی انسان که در هر کشور متفاوت است لحاظ نشده است .

**۲** – استاندارد های زیر شامل قوانینی می شوند که در این متن ، قوانین این قسمت از استاندارد ISO/IEC 14443 را تشکیل می دهد .

ISO/IEC 14443 کارت های شناسایی – کارتهای بدون تماس مدار مجتمع – کارتهای نزدیک

ISO/IEC 10373 کارت های شناسایی – روش های آزمون

ISO/IEC 7816-2 کارت های شناسایی – کارت های مجتمع شده با تماس

## ۳ قوانین و تعاریف

برای اهداف این استاندارد بین المللی، تعاریف داده شده در ISO/IEC 14443-1 و تعاریف زیر ارائه می شود :

### ۳.۱ مدت زمان بیت

زمانی که در طی آن یک حالت منطقی رخ می دهد ، در پایان آن یک بیت جدید شروع می شود.

### ۳.۲ کلید زنی شیفت فاز باینری

کلید زنی شیفت فاز است که در آن تغییر فاز 180 درجه است، که در نتیجه دوحالت فاز امکان دارد .

### ۳.۳ شاخص مدولاسیون

تعریف شده به عنوان [A - B] / [A + B] که در آن Aو B به ترتیب اوج و حداقل دامنه سیگنال.

### NRZ-L ۳.۴

روش کد گذاری بیتی است که در آن یک حالت منطقی در طول یک بیت که توسط یکی از دو حالت تعریف شده فیزیکی در رسانه های ارتباطی نشان داده است.

### ۳.۵ زیرحامل یا subcarrier

سیگنال RF تولید شده توسط مدولاسیون با فرکانس حامل Fc که با Fs نمایش داده می شود .

## ۴ اختصارات و نمادها

ASK Amplitude shift keying دامنه کلیدزنی شیفت

BPSK Binary phase shift keying کلید زنی شیفت فاز باینری

NRZ-L Non-return to zero, (L for level) بدون بازگشت به صفر

PCD Proximity coupling device دستگاه شامل کوپلینگ نزدیک ( ریدر )

PICC Proximity card کارت RFID

RF Radio frequency فرکانس رادیویی

Fc فرکانس کار (فرکانس حامل )

Fs فرکانس زیر حامل مدولاسیون

**۵ رابطه اولیه برای کارت ها**

رابطه اولیه بین PCDو PICC باید از طریق عملیات متوالی زیر انجام گیرد :

-فعال شدن PICC در محدوده فرکانس عملیاتی PCD

- PICC در حالت سکون منتظر یک دستور از PCD

-انتقال دستور توسط PCD

-انتقال پاسخ توسط PICC

این عملیات از قدرت RF و رابط سیگنال مشخص شده در بند های زیر استفاده می کند .

**۶ انتقال قدرت**

PCD باید یک میدان RF قوی تولید کند که به PICC متصل شود و قدرت را منتقل کند و باید برای ارتباط مدوله شده نیز باشد .

**۶.۱ فرکانس**

فرکانس (FC) باید در محدوده عملیاتی ۱۳،۵۶ مگاهرتز ± ۷ کیلو هرتز باشد .

**۶.۲ میدان عملیات**

حداقل میدان عملیات در حالت مدوله نشده باید Hmin باشد و مقدار A/m ۱.۵ (rms) باشد .

حداکثر میدان عملیات در حالت مدوله نشده باید Hmax باشد و مقدار A/m ۷.۵ (rms) باشد .

PICC باید بین Hmin و Hmax به طور پیوسته کار کند .

یک PCD باید یک میدان حداقل Hmin تولید کند که از حداکثر مقدار تعیین شده توسط کارخانه تجاوز نکند .

علاوه بر این، PCD باید قادر به تأمین انرژی هر PICC منفرد مرجع باشد (تعریف شده در روش های آزمون ) .

PCD نباید یک میدان بیشتر از مقدار مشخص شده در ISO/IEC 14443-1 (میدان مغناطیسی متناوب) در هر موقعیت ممکن PICC تولید کند .

روش های آزمون برای میدان عملیاتی PCD در ISO/IEC 10373 مشخص شده است .

**۷ سیگنال رابط**

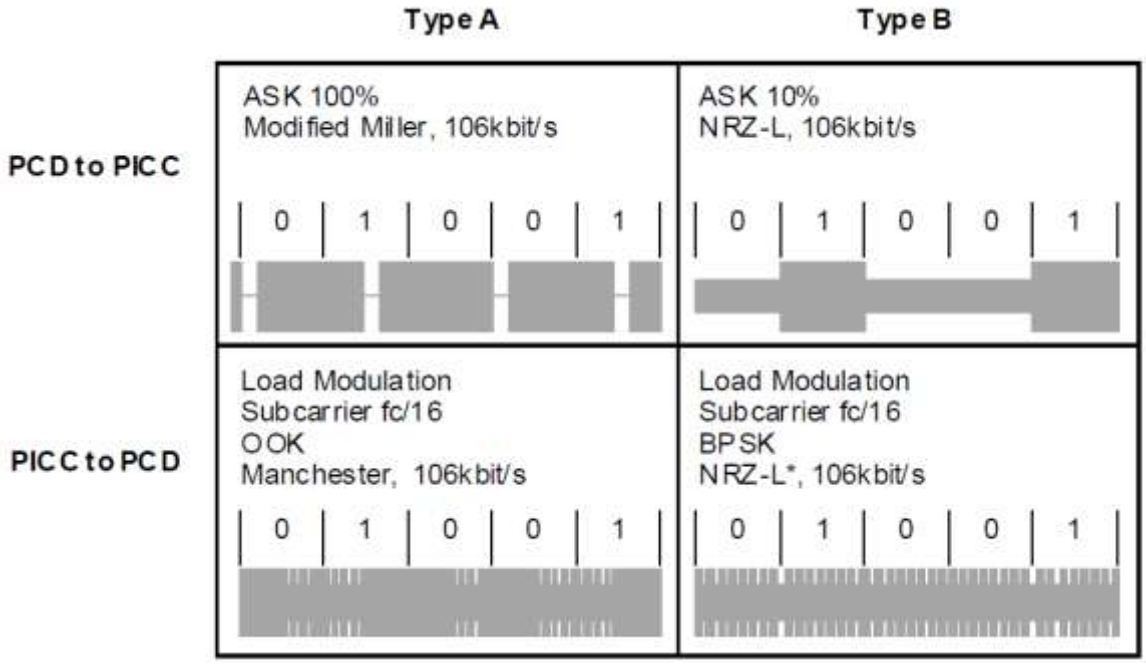
دو سیگنال ارتباطی نوع A و B ، در بند زیر توضیح داده شده است.

PCD باید بین روش های مدولاسیون در زمان بیکاری قبل از تشخیص حضور یک PICC نوع A یاB تغییر کند .

فقط یک سیگنال ارتباطی ممکن است در یک ارتباط فعال باشد تا زمانی که PCD غیر فعال کند یا PICC را از میدان خارج کنیم .

مراحل بعدی ممکن است از هر دو روش مدولاسیون استفاده می کند .

شکل ۳-۱، یک تصویر از مفاهیم شرح داده شده در بند های زیر می باشد .




شکل ۳ – ۱ نحوه ارتباط

# ۸ سیگنال ارتباطی نوع A

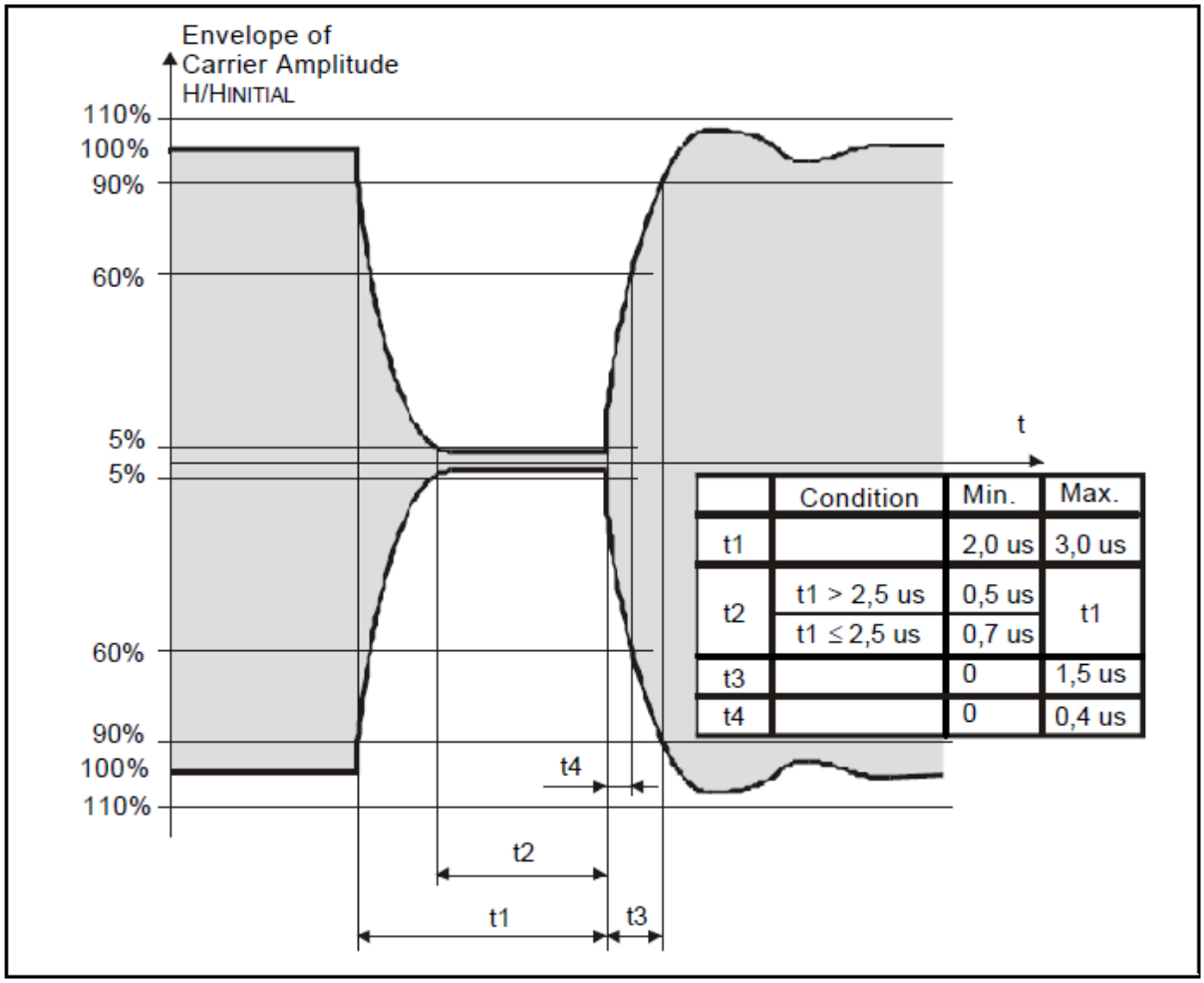
## ۸.۱ ارتباطات PCD به PICC

### ۸.۱.۱ سرعت داده

نرخ بیت داده برای انتقال در زمان نصب و ضدتداخل باید fc/128 باشد (~ 106 kbit/s) .

### ۸.۱.۲ مدولاسیون

ارتباط بین PCD و PICC با استفاده از اصول مدولاسیون ASK100٪ اتفاق می افتد و برای ایجاد یک مکث مانند شکل ۳-۲ عمل کنید .




شکل ۳-۲ : مکث در پروتکل نوع A

پوش میدان PCD باید به طور یکنواخت کاهش یابد تا به کمتر از ۵٪ از مقدار ابتدایی HINITIAL برسد و به مدت t2 کمتر از ۵٪ باقی بماند . این پوش باید بر شکل ۲ منطبق شود .

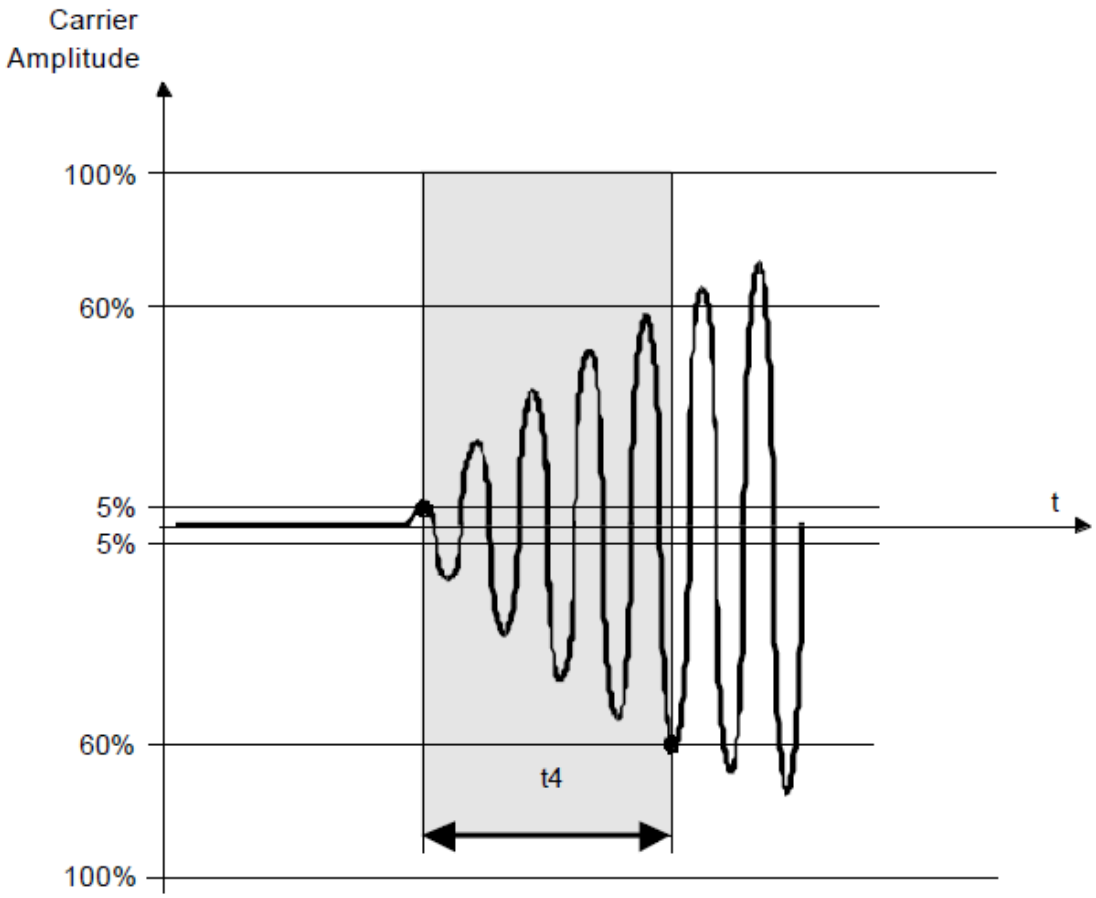
اگر پوش میدان PCD بطور یکنواخت کاهش نیابد ، زمان بین حداکثر محلی و زمان عبور همان مقدار ، قبل از حداکثر محلی نباید از 0.5 uS تجاوز کند . این باید تنها در صورتی که حداکثر محلی بیشتر از ۵٪ از HINITIAL باشد اعمال می شود.

جهش ها باید بین ۹۰٪ و 110٪ از HINITIAL باقی بماند .

PICC باید " پایان هر مکث " را بعد از اینکه میدان به ۵٪ از HINITIAL و قبل از رسیدن به ۶۰ ٪ آن تشخیص دهد .

توجه : در سیستم طراحی شده ای که تنها با یک کارت در یک زمان سر و کار دارد ، T4 مهم نیست .

شکل ۳-۳ : تعیین پایان یک مکث



### ۸.۱.۳ کدگذاری و نمایش بیت

حالات زیر تعریف شده اند :

حالت X : بعد از یک زمان 64/Fc یک مکث باید اتفاق بیفتد .

حالت Y : در مدت زمان یک بیت (128/Fc) هیچ مدولاسیونی نباید اتفاق بیفتد .

حالت Z : در شروع یک بیت یک مکث باید رخ دهد .

حالات قبل برای کدگذاری اطلاعات زیر به کار گرفته می شوند :

سطح منطقی یک : حالت X

سطح منطقی صفر : حالت Y به غیر از دو حالت زیر :

۱ ) اگر دو صفر یا بیشتر در کنار یکدیگر قرار بگیرند ، حالت Z باید برای صفر دوم به بعد استفاده شود .

۲ ) اگر اولین بیت بعد از شروع فریم صفر باشد ، حالت Z باید نمایش صفرهای متوالی بعد از آن استفاده شود .

شروع ارتباط : حالت Z

پایان ارتباط : منطق صفر به همراه حالت Y

بدون اطلاعات : حداقل دو حالت Y

## ۸.۲ ارتباط PICC با PCD

### ۸.۲.۱ نرخ داده

نرخ بیت داده برای انتقال در زمان نصب و ضدتداخل باید fc/128 باشد (~ 106 kbit/s) .

### ۸.۲.۲ بار مدولاسیون

PICC باید قادر به برقراری ارتباط با PCD از طریق محیط کوپلینگ القایی باشد جایی که فرکانس کریر بارگذاری می شود تا فرکانس زیر حامل Fs را تولید کند . ساب کریر باید توسط سوییچینگ یک بار در PICC تولید شود .

دامنه مدولاسیون بار باید حداقل 30/H1.2 mV ( پیک ) باشد ، جایی که H مقدار موثر قدرت میدان مغناطیسی بر حسب A/M می باشد .

روش های آزمون مدولاسیون بار PICC در استاندارد بین المللی ISO/IEC 10373 تعریف شده است.

### ۸.۲.۳ زیر حامل یا ساب کریر

FS فرکانس حامل باید fc/16 (۸۴۷ هرتز). در نتیجه، در مقدار دهی اولیه و anticollision، مدت زمان یک بیت است که معادل ۸ دوره از حامل است.

### ۸.۲.۴ مدولاسیون ساب کریر یا زیر حامل

هر دوره تناوب بیت با فاز تعریف شده مربوط به زیرحامل شروع می شود . دوره تناوب بیت با حالت بارگذاری شده ساب کریر شروع می شود .

زیرحامل باید با استفاده از کلید زنی روشن / خاموش با توالی تعریف شده در .8.2.5 مدوله شوند .

### ۸.۲.۵ کدگذاری و نمایش بیت

کدگذاری بیت باید منچستر با تعاریف زیر باشد :

| | |
|---|---|
| sequence D | the carrier shall be modulated with the subcarrier for the first half (50%) of the bit duration |
| sequence E | the carrier shall be modulated with the subcarrier for the second half (50%) of the bit duration |
| sequence F | the carrier is not modulated with the subcarrier for one bit duration |
| logical "1" | sequence D |
| logical "0" | sequence E |
| Start of communication | sequence D |
| End of communication | sequence F |
| No information | no subcarrier |

## ۹ سیگنال رابط نوع B

### ۹.۱ ارتباط PCD به PICC

#### ۹.۱.۱ نرخ داده

نرخ بیت داده برای انتقال در طول نصب اولیه و ضدتداخل باید به طور اسمی fc/128 (~ ۱۰۶ کیلوبیت / ثانیه) باشد. مرزهای تحمل و کمی در ISO/IEC 14443-3 تعریف شده است.

#### ۹.۱.۲ مدولاسیون

ارتباط بین PCD و PICC از طریق مدولاسیون دامنه ASK 10% انجام می گیرد .

شاخص مدولاسیون باید حداقل ۸ درصد و حداکثر ۱۴ درصد باشد.

موج مدولاسیون باید مطابق با شکل ۴ باشد . لبه افزایشی و کاهشی مدولاسیون باید یکنواخت باشد.

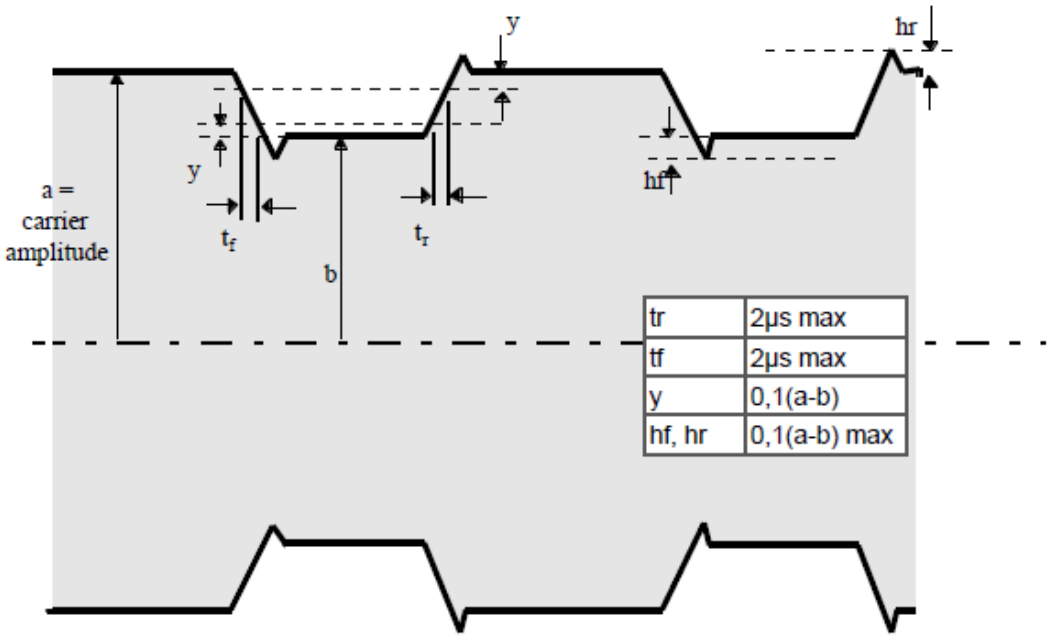



**Figure 4 -- Type B modulation waveform**

### ۹.۱.۳ کدگذاری و نمایش بیت

فرم کدگذاری بیت باید به صورت NRZ-L با سطوح منطقی مشخص شده در زیر باشد :

سطح "۱" مقدار بالای میدان کریر

سطح "۰" مقدار پایین میدان کریر

## ۹.۲ ارتباط PICC با PCD

### ۹.۲.۱ نرخ داده

نرخ بیت داده برای انتقال در طول نصب اولیه و ضدتداخل باید به طور اسمی fc/128 (~ ۱۰۶ کیلوبیت / ثانیه) باشد.

### ۹.۲.۲ بار مدولاسیون

PICC باید قادر برقراری ارتباط با PCD از طریق منطقه کوپلینگ القایی باشد که در آن انرژی میدان برای تولید یک ساب کریر با فرکانس Fs بارگذاری می شود . ساب کریر باید توسط سوییچینگ یک بار در PICC تولید شود .

دامنه مدولاسیون بار باید حداقل 30/H1.2 mV ( پیک ) باشد ، جایی که H مقدار موثر قدرت میدان مغناطیسی بر حسب A/M می باشد .

روش های آزمون مدولاسیون بار PICC در استاندارد بین المللی ISO/IEC 10373 تعریف شده است.

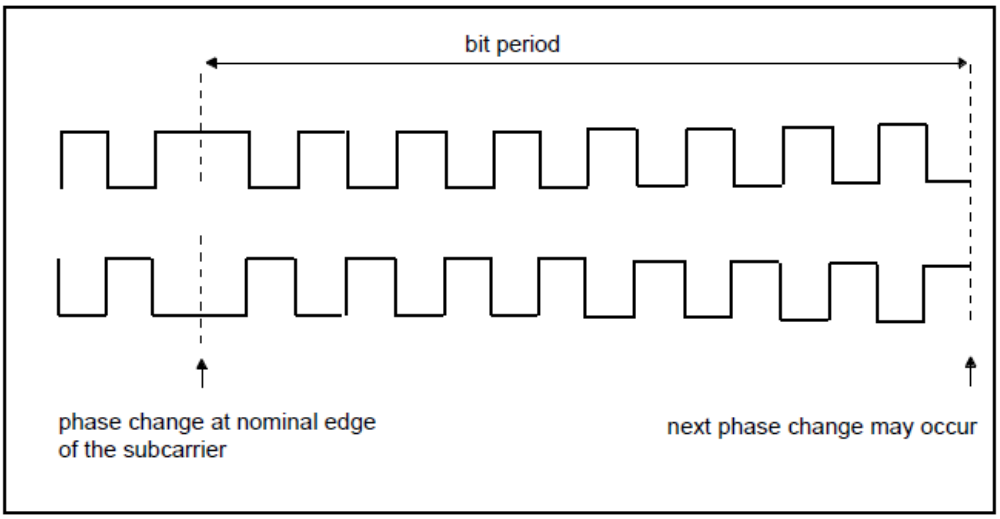
### ۹.۲.۳ زیر حامل

فرکانس FS باید fc/16 (~۸۴۷ کیلوهرتز) باشد . در نتیجه ، در مقدار دهی اولیه و ضدتداخل ، مدت زمان یک بیت برابر است با ۸ دوره از زیر حامل .

PICC باید فقط زمانی ساب کریر تولید کند که قصد ارسال دیتا داشته باشیم .

### ۹.۲.۴ مدولاسیون ساب کریر

ساب کریر باید به صورت BPSK مدوله شده باشد ، شیفت فاز باید فقط در مکان های فراز و فرود ساب کریر اتفاق بیفتد .



### ۹.۲.۵ کدگذاری و نمایش بیت

کدگذاری بیت باید به صورت NRZ-L جایی که یک تغییر سطح منطقی باید با تغییر فاز180 درجه ساب کریر اعمال شود .

سطوح منطقی اولیه برای NRZ-L در آغاز یک فریم PICC باید با مراحل زیر شروع شود :

- پس از هر فرمان از PCD زمان گارد TR0 باید طوری اعمال شود که در آن PICC نباید ساب کریر تولید کند . TR0 باید بزرگتر از 64/fs باشد.

- PICC سپس باید یک ساب کریر بدون انتقال فاز برای همزمان کردن TR1 ایجاد کند . این کار یک ساب کریر با فاز مرجع Ø0 تولید می کند . TR1باید بیشتر از 80/fs باشد .

- این حالت فاز اولیه Ø0 ساب کریر باید به صورت سطح منطقی "1" تعریف شده به طوری که در مرحله اول انتقال نشان دهنده یک تغییر منطقی از "1" به "۰" باشد .

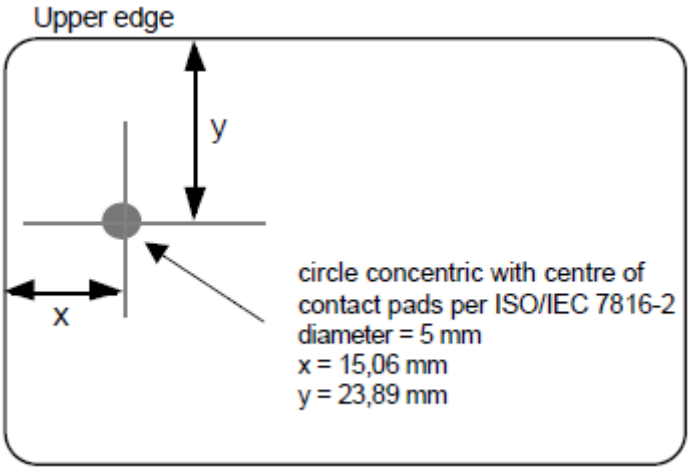
- پس از آن سطح منطقی باید با توجه به فاز مرجع ساب کریر تعریف شود :

Ø0 : نمایش دهنده سطح "۱"

Ø0 + ۱۸۰ : نمایش دهنده سطح "۰"

## ۱۰ حداقل محدوده کوپلینگ PICC

آنتن کوپلینگ PICC ممکن است به هر شکل و هر جایی باشد اما باید منطقه نشان داده شده در شکل ۶ را مشتمل شود .




شکل ۳-۴ : حداقل محدوده کوپلینگ PICC

لبه های بالا و چپی در ISO/IEC 7816-2 مشخص شده اند . منطقه سایه دار دارای قطر ۵.۰ میلیمتر می باشد .

# استاندارد 14443-3

## کارت های شناسایی – مدارات مجتمع کارت های بدون تماس – کارت های نزدیک

## قسمت سوم : راه اندازی و ضدتداخل

# ۱ – هدف

در این قسمت از ISO/IEC14443 مسائل زیر توضیح داده می شود :

- رای گیری PICC هایی که داخل میدان PCD می شوند.

- قالب بایت، فریم و زمانبندی استفاده شده در زمان فاز ابتدایی ارتباط بین PICC,PCD

- محتویات دستورات ATQ,REQ ابتدایی

- روشهایی جهت تشخیص و تماس با یک کارت در میان چندین کارت

- دیگر پارامترهای مورد نیاز برای برقراری ارتباط بین PICC,PCD

- وسایل اختیاری برای آسان کردن و سرعت بخشیدن به انتخاب یک کارت در میان چندین کارت.

پروتکل ها و دستورات استفاده شده در لایه های بالاتر و کاربردهایی که در بعد از فاز ابتدایی استفاده می شود در ISO/IEC 14443-4 توضیح داده می شود.

**۲ –** استاندارد های زیر شامل قوانینی می شوند که در این متن ، قوانین این قسمت از استاندارد ISO/IEC 14443 را تشکیل می دهد .

ISO/IEC 14443 کارت های شناسایی – کارتهای بدون تماس مدار مجتمع – کارتهای نزدیک

ISO/IEC ۱۰۳۷۳ کارت های شناسایی – روش های آزمون

ISO/IEC 7816-2 کارت های شناسایی – کارت های مجتمع شده با تماس

**۳- عبارات و معانی**

برای اهداف این استاندارد بین المللی، عبارات و معانی ارائه شده در ISO/IEC 7816-3, ISO/IEC 14443-2 مورد نیاز است.

**۳-۱- حلقه ضد تداخل**

الگوریتم استفاده شده برای مهیا کرده یک ارتباط بین PCD و یک و چندین PICC درون میدان

### ۳-۲- پروتکل تشخیص تداخل بیت

روش ضد تداخل با به کارگیری تشخیص تداخل از روی سطح بیت درون فریم می باشد. یک تداخل زمانی رخ می دهد که دو PICC الگوی مکمل بیتی خود را به PCD ارسال کنند. در این حالت الگوهای بیت با هم مخطوط شده و «کریر» با «ساب کریر» در تمام مدت بیت مدوله می شوند.

PCD بیت های تداخل یافته را تشخیص و با روش Cascadeی (متوالی) ID همه ی PICC ها را شناسایی می کند.

### ۳-۳- بایت

یک بایت شامل ۸ بیت اطلاعات که با نمادهای b1 تا b8 نشان داده می شوند، می باشد که از با ارزش ترین بیت b8 تا کم ارزش ترین b1 قرار گرفته اند.

### ۳-۴- تداخل

ارسال همزمان دو PICC به یک PCD به طوری که PCD قادر به تشخیص منبع اطلاعات

### ۳-۵- واحد زمان اصلی (etu)

در این قسمت از ISO/IEC 14443، یک etu برابر : 1etu=128/fc

### ۳-۶- فریم

یک فریم یک سری از بیت های داده و بیتهای اختیاری خطا به همراه حائل هایی در ابتدا و انتها می باشد.

### ۳-۷- لایه بالاتر

متعلق به کاربرد یا پروتکل لایه بالاتر می باشد که در این قسمت توضیح داده نشده است.

### ۳-۸- پروتکل اسلات زمانی

روشی که در آن یک PCD کانال های منطقی با یک یا چند PICC تشکیل می دهد. کـه از اسـلات زمـانی برای پاسخ PICC استفاده می کند.

### ۳-۹- شناسه منحصر به فرد UID

UID یک شماره که برای الگوریتم نوع A لازم است، می باشد.

### ۴ – نماد ها

| | |
|---|---|
| ATA | Answer To Attrib |
| ATQ | Answer To Request |
| ATQA | Answer To Request of Type A |
| ATTRIB PICC | selection command |
| BCC UID CLn | checkbyte, calculated as exclusive-or over the 4 previous bytes |
| CLn | Cascade level n, $3 \geq n \geq 1$ |
| CT | Cascade Tag, ‘88’ |
| CRC_A | Cyclic Redundancy Check error detection code defined in 6.1.10 |
| DESEL | Deselect Command |
| E | End of communication, Type A |
| EGT | Extra Guard Time |
| EOF | End Of Frame, Type B |
| etu | elementary time unit. Duration of 1 bit of data transmission. |
| FGT | Frame Guard Time |
| fc | carrier frequency (13,56 MHz) |
| fs | subcarrier frequency |
| ID | IDentification number |

INF INFormation field belonging to higher layer

LSB Least Significant Bit

MSB Most Significant Bit

N Number of anticollision slots or PICC response probability in each slot

n Variable integer value as defined in the specific clause

NAD Node ADdress

NVB Number of Valid Bits

P Odd parity bit

PARAM Parameter in Attribute format

PCD Proximity Coupling Device (Reader)

PICC Proximity Card

PUPI Pseudo-Unique PICC Identifier

R Slot number chosen by the PICC during the anticollision sequence

REQA Request Command, Type A

RFU Reserved for Future ISO/IEC Use

S Start of communication, Type A

SAK Select AcKnowledge

SEL Select Command

TR0 Minimum delay of silence between PCD off and PICC

TR1 Minimum subcarrier on duration before PICC data transmission.
UID Unique IDentification

UID*n* Byte number *n* of unique identification, $n \geq 0$

**۵- رای گیری**

زمانی که یک PICC در مقابل یک میدان مدوله نشده قرار می گیرد، باید قادر به پذیرفتن درخواست در ۵ میلی ثانیه باشد.

برای تشخیص PICC هایی که به میدان انرژی وارد می شود، یک PCD دستورهای درخواست متوالی ارسال می کند و منتظر یک ATQ می ماند. دستورات درخواست در هر قسمتی باید از REQA توضیح داده شده در اینجا استفاده کنند و بعلاوه ممکم است از دیگر روشهای کدگذاری در ضمیمه استفاده کند. این رویه را راگیری می نامند.

**۶- نوع A – نصب و ضد تداخل**

این بخش پروتکل تشخیص تداخل بیت قابل اجرا برای PICC نوع A را توصیف می کند.

**۶-۱- بایت، فریم و قالب دستور و زمان بندی**

این بخش بایت، فریم و قالبهای دستور و زمانبندی استفاده شده در طول ارتباط، پیکربندی و ضد تداخل مشخص می کند. برای نمایش و کد گذاری بیت به ISO/IEC 14443-2 مراجعه کنید.

**۶-۱-۱- زمان تاخیر فریم**

زمان تاخیر فرمی FDT، زمان بین دو فریم ارسالی در جهات مخالف هم تعریف می شود.

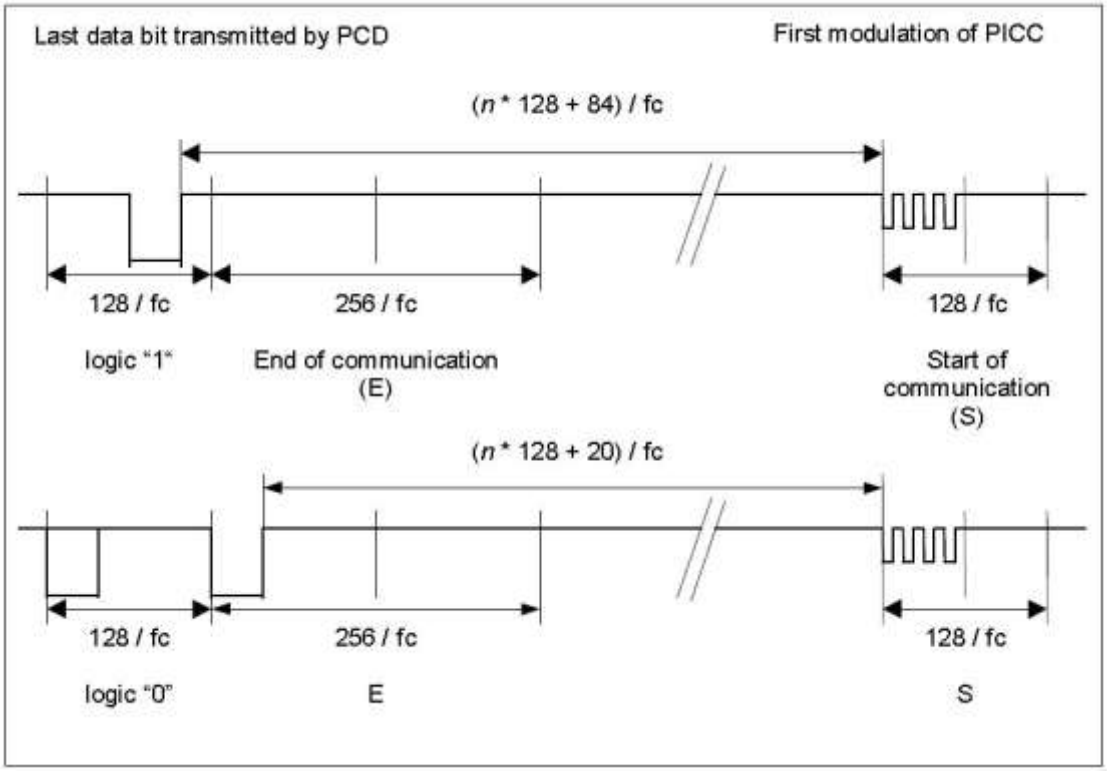
**۶-۱-۲- زمان محافظ فریم**

زمان محافظ فریم FGT، حداقل زمان تاخیر فریم می باشد.

**۶-۱-۳- زمان تاخیر فریم از PCD به PICC**

این زمانی است که بین آخرین مکث ارسالی توسط PCP و اولین لبه مدولاسیون همراه بیت شروع ارسالی توسط PICC می باشد و باید مطابق زمان بندی معین شده در شکل ۳-۳-۱ باشد، جایی که n یک مقدار عدد صحیح است.

شکل ۳-۳-۱ : زمان تاخیر فریم PICC به PCD




جدول۳-۳-۱ مقادیر n و FDT وابسته به نوع دستور و حالت منطقی آخرین بیت داده ارسالی در این دستور مشخص می کند.

جدول ۳-۳-۱ : زمان تاخیر فریم PICC به PCD

| Command type | n (integer value) | FDT | |
|---|---|---|---|
| | | last bit = (1)b | last bit = (0)b |
| REQA Command<br>WAKE-UP Command<br>ANTICOLLISION Command<br>SELECT Command | 9 | 1236 / fc | 1172 / fc |
| All other commands | ≥ 9 | (n * 128 + 84) / fc | (n * 128 + 20) / fc |

تذکر : مقدار n=9 یعنی همه ی PICC های درون میدان باید به طور هماهنگ که برای ضد تداخل ضروری است، پاسخ دهند.

برای دیگر دستورات PICC باید از اولین لبه مدولاسیون به همراه بیت شروع می باشد، اطمینان حاصل نماید.

### ۶-۱-۴- زمان تاخیر فریم از PICC به PCD

این زمان بین آخرین مدولاسیون ارسالی توسط PICC و اولین مکث ارسالی توسط PCD مـی باشـد و بایـد حداقل $117\ {}^{2}/_{fc}$ باشد.

### ۶-۱-۵- زمان محافظ در خواست

زمان محافظ درخواست حداقل زمان بین بیتهای شروع دو دستور درخواست متـوالی مـی‌باشـد. کـه مقـدار $7000/_{fc}$ می باشد.

### ۶-۱-۶- قالبهای فریم

در ادامه انواع فریم برای پروتکل تشخیص تداخل بین مشخص شده است.

### ۶-۱-۷- دستورات WAKE-UP,REQA

فریم های درخواست و wake-up برای آغاز ارتباط استفاده می شوند و شامل ترتیب زیر می باشد.

* شروع ارتباط

* ۷ بیت ارسـالی کـه اولـین آن LSB اسـت. (محتویـات اطلاعـات بـرای اسـتاندارد REQA «۲۶» و بـرای درخواست wakeup «۵۲» می باشد.)

* پایان ارتباط

بیت توازن اضافه نشده است.

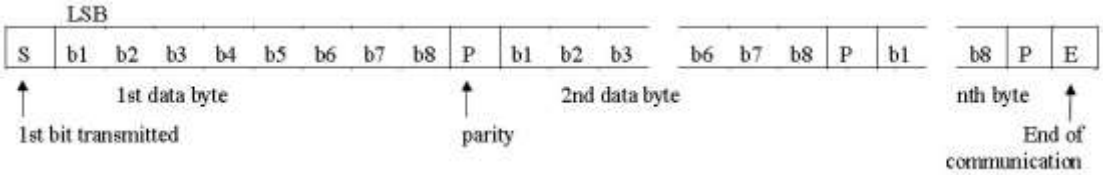
### ۶-۱-۸- فریم استاندارد

فریم های استاندارد برای جابجایی داده ها استفاده می شوند و شامل :

* شروع ارتباط

* (۸ بیت داده + بیت توازن فرد) * n : برای $n \geq 1$ LSB هربایت داده ای اول ارسال می‌شود. هر بایت داده به همراه یک بیت توازن فرد می باشد.

* پایان ارتباط




شکل ۳-۳-۲ : فریم استاندارد

### ۶-۱-۹- فریم ضد تداخل بیت گرا

یک تداخل زمانی به وقوع می پیوندد که حداقل دو PICC دو الگوی بیت متفاوت را به PCD ارسال کنند. در این حالت کریر با سب کریر در تمام مدت بیت مدوله می شوند.

فریم های ضد تداخل بیت گرا فقط در زمان حلقه ضد تداخل استفاده می شوند و در حقیقت فریم های استانداردی می باشند که به طول ۷ بایت داده و به دو قسمت تقسیم می شوند : قسمت ۱ برای انتقال از PCD به PICC و قسمت ۲ برای انتقال از PICC به PCD.

برای طول های قسمت ۱ و ۲، قوانین زیر باید اعمال شود :

قانون اول : جمع بیتهای داده باید ۵۶ شود.

قانون دوم : حداقل طول قسمت ۱ باید ۱۶ بیت داده باشد.

قانون سوم : حداکثر طول قسمت ۱ باید ۵۵ بیت داده باشد.

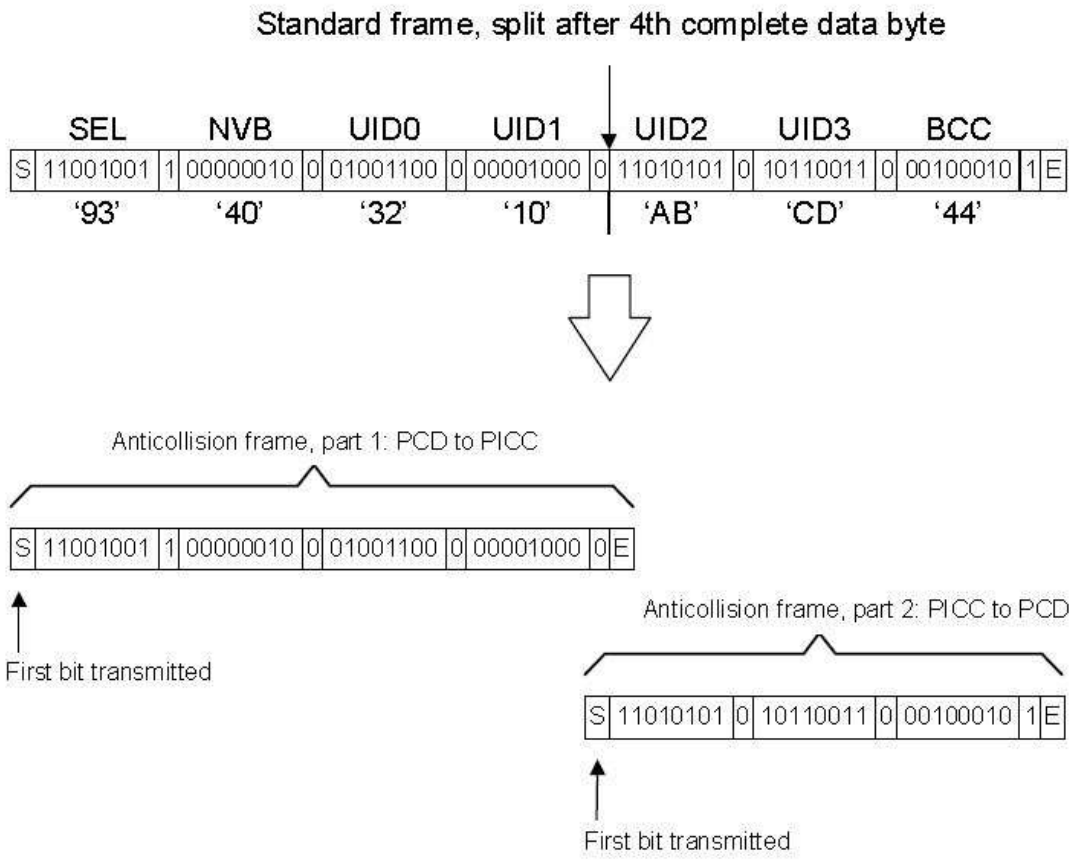
در نتیجه، حداقل طول قسمت ۲ باید یک بیت داده و حداکثر طول آن باید ۴۰ بیت داده باشد.

از آنجایی که جدایی می تواند در هر بیتی در یک بایت داده اتفاق بیفتد، دو حالت تعریف می‌شود :

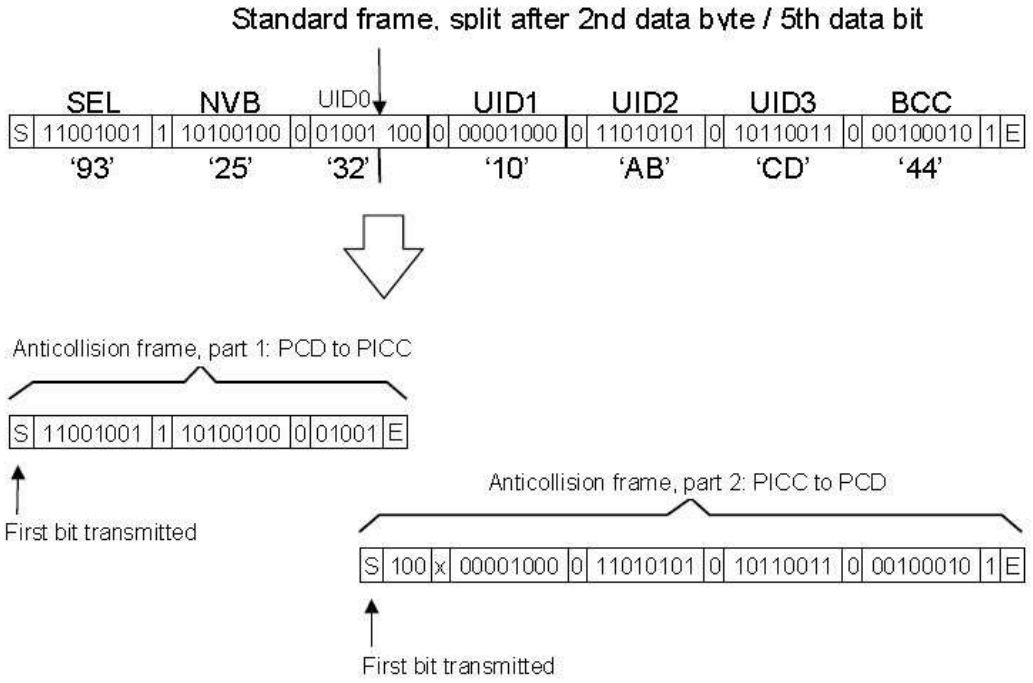
حالت FULL BYTE : جدایی بعد از یک بابت داده کامل باشد. یک بیت توازن بعد از آخرین بیت داده قسمت اول اضافه می شود.

حالت SPLIT BYTE : جدایی درون یک بایت داده رخ دهد، هیچ بیت توازنی بعد از آخرین بیت داده قسمت اول اضافه نمی شود.

مثالهای زیر برای حالتهای SPLIT BYTE, FULL BYTE شکل و ترتیب انتقال بیت را مشخص می کند.




شکل ۳-۳-۳ : ساماندهی و انتقال بیت در فریم ضدتداخل بیت گرا ، FULL BYTE CASE




شکل ۳-۳-۴ : ساماندهی و انتقال بیت در فریم ضدتداخل بیت گرا ، SPLIT BYTE CASE

برای یک SPLIT BYTE ، اولین بیت توازن قسمت دوم باید نادیده گرفته شود.

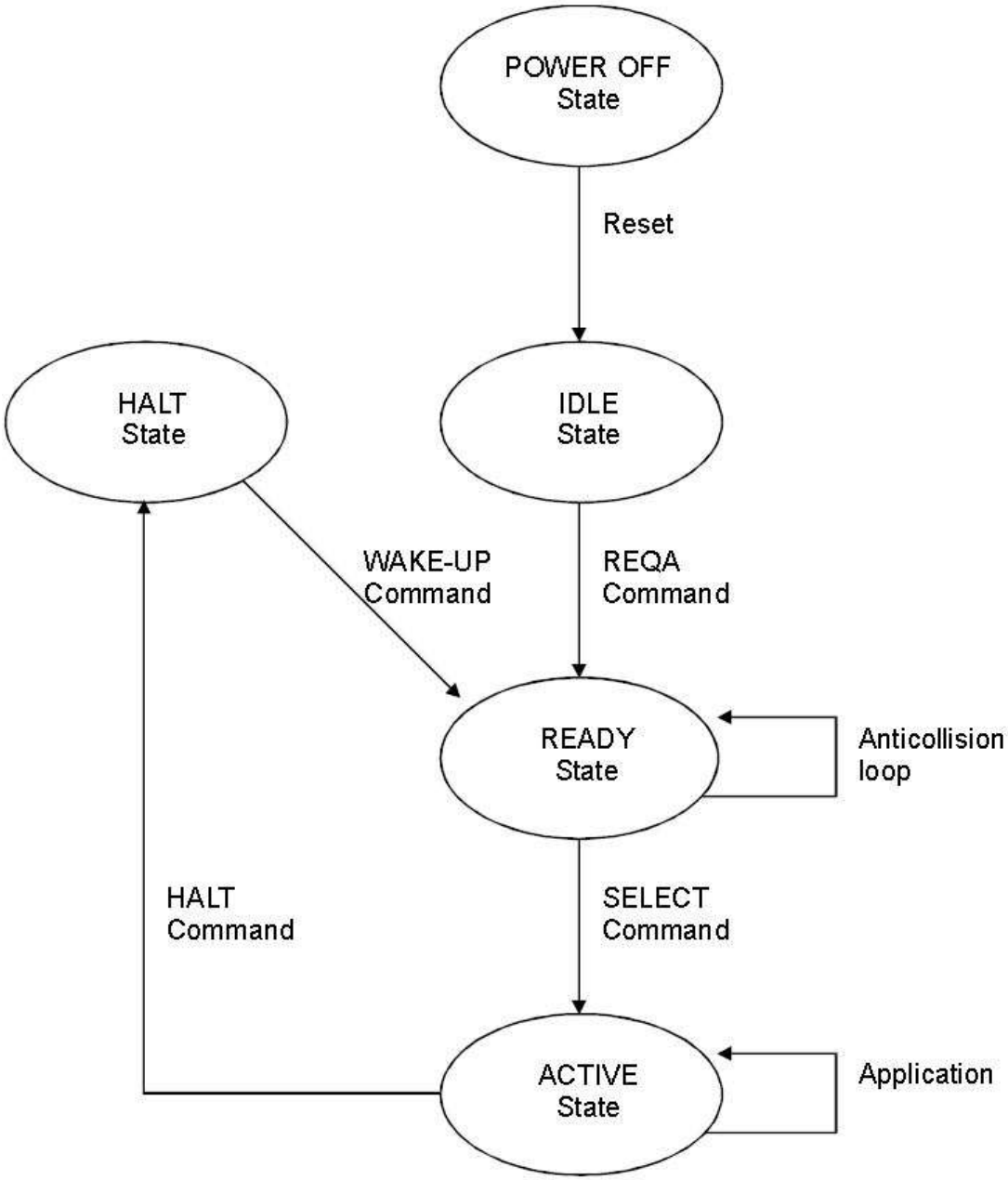
### ۶-۱-۱۰- CRC-A

روند کدگذاری و چک کردن CRC-A در ITU-T مشخص شده است. چند جمله ای مولد برای تولید بیتهای بررسی ..... می باشد.

مقدار اولیه باید «6363» باشد. CRC-A باید به بایتهای داده اضافه شود و توسط فریم ه‌ای استاندارد ارسال شود.

## ۶-۲- حالتهای PICC

بخش های زیر توضیحاتی راجع به حالتهای یک PICC نوع A خاص برای پروتکل تشخیص تداخل بیت داده می شود.




شکل ۳-۳-۵ : دیاگرام حالت های PICC نوع A

### ۶-۲-۱- حالت POWER-OFF

در این حالت، PICC به دلیل نبود انرژی کافی کریر نمی تواند حساب کریر تولید و منتشر کند.

### ۶-۲-۲- حالت IDLE

بعد از اینکه میدان به مدت حداکثر تاخیر مشخص شده در بند ۵ فعال شد، PICC باید بـه ایـن حالـت وارد شود. در این حالت PICC روشن است و قادر به آشکار سـازی و تشـخیص دسـتورات REQA و wake-up معتبر از سوی PCD می باشد.

### ۶-۲-۳- حالت READY

یک PICC زمانی به این حالت وارد می شود که یک پیام معتبر REQA یا wake-up دریافت شود و زمانی که UID آن شناسایی و انتخاب شود از این حالت خارج می شود. در این حالت روش ضد تداخل فریم بیـت یا هر روش ضد تداخل دیگری می تواند اعمال شود. مراحـل Cascade بـرای بدسـت آوردن UID در ایـن حالت به کار گرفته می شوند.

### ۶-۲-۴- حالت ACTIVE

زمانی به این حالت وارد می شود که PICC توسط UID آن انتخاب شود.

### ۶-۲-۵- حالت HALT

زمانی وارد این حالت می شود که یا دستورHALT توضیح داده شده در ۶-۳-۴ یا توسط یک دستور خـاص یک کاربرد که در این بخش از ISO/IEC14443 تعیین نشده اجرا شود. در این حالت یک PICC باید فقط به دستور WAKE-UP جواب بدها، که PICC را به حالت READY منتقل می کند.

تذکر : PICC ها که در حالت HALT هستند در هیچ یک از ارتباطات بعدی شرکت نخواهند کرد مگر اینکه دستور WAKE-UP اعمال شود.

### ۶-۳- دستورات

دستوراتی که توسط PCD برای مدیریت ارتباط بین چند PICC استفاده می شوند، عبارتند از :

* PEQA
* WAKE-UP
* ANTICOLLISION
* SELECT
* HALT

این دستورات از قالبهای فریم و بایت توضیح داده شده استفاده می کند.

### ۶-۳-۱- دستور REQA

این دستور توسط PCD گشتن PICC های نوع A درون میدان ارسال می شود.

### ۶-۳-۲- دستور WAKE-UP

این دستور برگرداندن PICC هایی که به حالت HALT وارد شدند تا به حالت READY دوباره باز گردند توسط PCD ارسال می شود. آنها باید در روندهای ضد تداخل و انتخاب آینده شرکت نمایند.

جدول ۳-۳-۲ کدگذاری WAKE-UP,REQA که از قالب فریم درخواست استفاده می‌کنند، نشان می دهد.

جدول ۳-۳-۲ : کدگذاری فریم درخواست

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | Meaning |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | '26' = REQA |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | '52' = WAKE-UP |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | '35' = Optional time slot method, see Annex C |
| 1 | 0 | 0 | x | x | x | x | '40' to '4F' = Proprietary |
| 1 | 1 | 1 | 1 | x | x | x | '78' to '7F' = Proprietary |
| all other | | | | | | | RFU |

### ۶-۳-۳- دستور ANTICOLLISION و دستور SELECT

این دستورات در مدت یک حلقه ضد تداخل استفاده می شوند. دستورات ANTICOLLISION و SELECT شامل :

* کد انتخاب SEL (۱ بایت)

* تعداد بیتهای معتبر NVB (یک بایت)

* ۰ تا ۴۰ بیت داده UID CLn مطابق با مقدار NVB. SEL مرحله Cascade (CLn) را مشخص می کند.

NVB تعداد بیتهای معتبر را VIPCLn ارسالی توسط PCD را مشخص می کند.

تذکر : تا زمانی که NVB ۴۰ معتبر را نشان ندهد، دستور ضد تداخل نامیده می شود، جایی که PICC در حالت READY باقی می ماند. اگر NVB ۴۰ بیت داده UID CLn را نشان دهد ("NVB="70)، یک CRC-A باید ضمیمه شود. این دستور SELECT نامیده می شود اگر UID PICC کامل خود را ارسال کرده باشد، از حالت READY به ACTIVE می رود و در پاسخ SAK خود نشان می دهد که UID کامل است. در غیر این صورت، PICC در حالت READY باقی مانده و PCD باید یک حلقه ضد تداخل جدید با مرحله افزایش یافته Cascadeی را اندازی کند.

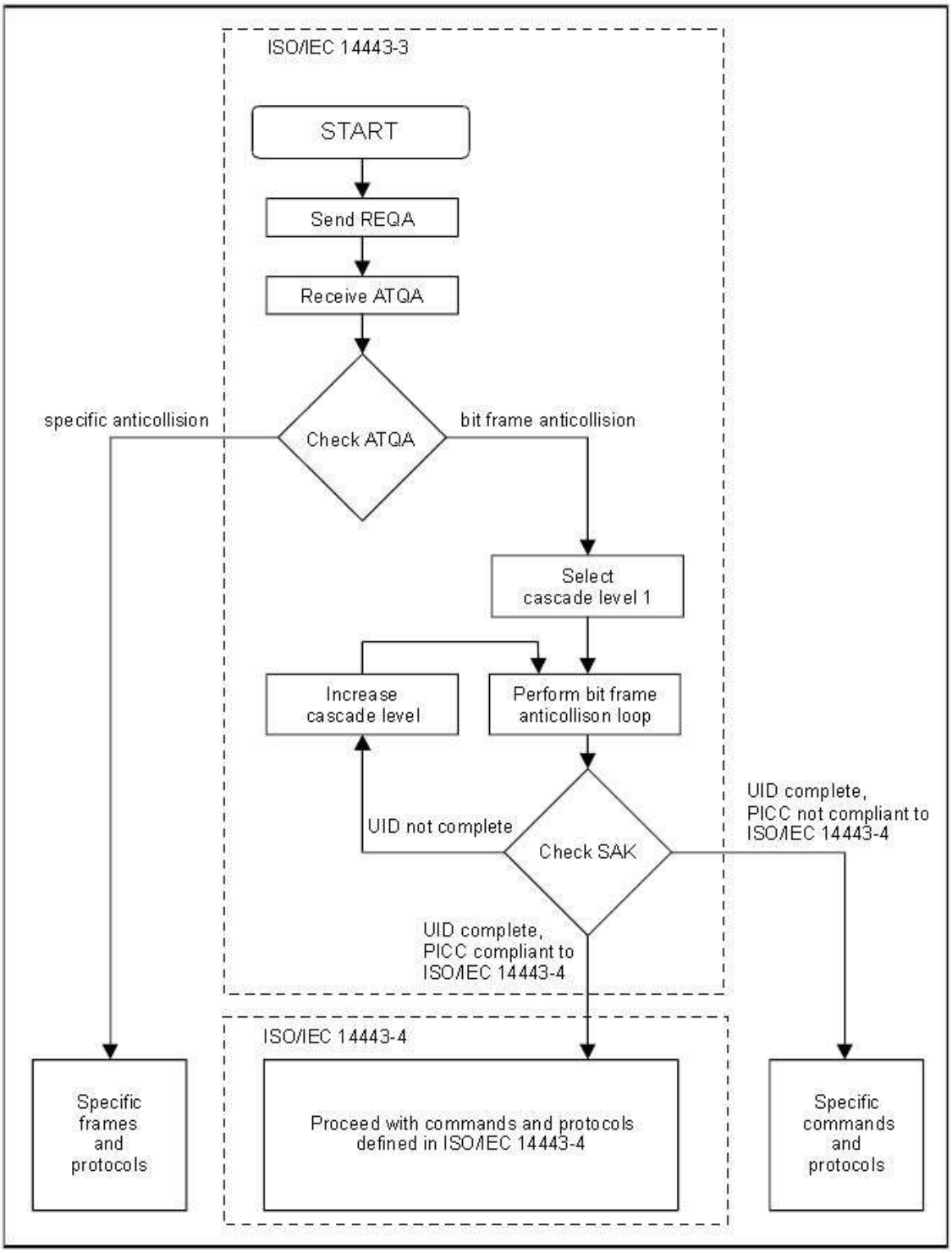
### ۶-۳-۴- دستور HALT

این دستور شامل ۴ بایت می شود که باید به صورت استاندارد فریم ارسال شود. اگر پاسخ‌های PICC ها با هر مدولاسیونی در مدت یک میلی ثانیه بعد از پایان فریم HALT باشد، این پاسخ ها باید به عنوان عدم اعتبار معنی شوند.

### ۶-۴- مراحل انتخاب

هدف از مراحل انتخاب بدست آوردن UID یک PICC و انتخاب این PICC برای ارتباطات بعدی می باشد .

۶-۴-۱ فلوچارت مراحل انتخاب




شکل ۳-۳-۶ : فلوچارت راه اندازی و ضد تداخل برای PCD

### ۶-۴-۲- پاسخ به درخواست ATQA

بعد از دستور در خواست (REQA) توسط PCD ارسال شد، همه PICC ها به طور همزمان ATQA هایشان را ارسال می کنند که به صورت ضد تداخل مناسب در دو بایت داده کد گذاری شده اند.

اگر چندین پاسخ PICC موجود باشد، تداخل ممکن است اتفاق بیفتد. PCD باید یک تداخل را درون ATQA به عنوان b (۱) رمزگشایی کند. این نتیجه می دهد که همه ی ATQA ها با هم OR منطقی شده اند.

### ۶-۴-۲-۱- رمزنگاری ATQA

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

جدول ۳-۳-۳ : رمزنگاری ATQA

۶-۴-۲-۲- قوانین رمزنگاری برای فریم ضد تداخل بیت

قانون ۱ : بیتهای b8,b7 نشان دهنده اندازه UID می باشند.

قانون ۲ : یکی از پنج بیت b5,b4,b3,b2,b1 باید b (۱) باشد تا نشان دهنده فریم ضد تداخل بیت باشد.

جدول ۳-۳-۴ : حالت های b7 و b8

| b5 | b4 | b3 | b2 | b1 | Meaning |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | bit frame anticollision |
| 0 | 1 | 0 | 0 | 0 | bit frame anticollision |
| 0 | 0 | 1 | 0 | 0 | bit frame anticollision |
| 0 | 0 | 0 | 1 | 0 | bit frame anticollision |
| 0 | 0 | 0 | 0 | 1 | bit frame anticollision |
| All other values | | | | | RFU |

جدول ۳-۳-۵ : حالتهای b5 تـــا b1

### ۶-۴-۳- ضد تداخل و انتخاب

#### ۶-۴-۳-۱- حلقه ضد تداخل به همراه مرحله Cascade

الگوریتم زیر باید در حلقه ضد تداخل اعمال شود :

گام ۱ : PCD باید SEL به همراه کد مربوط به نوع ضد تداخل انتخابی و مرحله Cascade اختصاص می دهد.

گام دوم : PCD باید NVB با مقدار «۲۰» را مشخص کند.

تذکر : این مقدار مشخص می کند که PCD هیچ قسمت از UID خود را ارسال نمی کند. در نتیجه این دستور همه PICC های درون میدان را مجبور به ارسال UID CLN کامل خود می کند.

گام ۳ : PCD، SEL و NVB را ارسال می کند.

گام ۴ : همه ی PICC های درون میدان باید UID CLn کامل خود را ارسال کنند.

گام ۵ : با این فرض که PICC های درون میدان شماره سریال منحصر به فرد دارند، در این زمان اگر بیشتر از یک PICC پاسخ دهد، یک تداخل رخ می دهد. اگر هیچ تداخلی ندهد، گام های ۶ تا ۱۰ باید نادیده گرفته شوند.

گام ۶ : PCD باید مکان اولین تداخل را شناسایی کند.

گام ۷ : PCD مقدار NVB را طوری تعیین کند که تعداد بیتها معتبر UID CLn را مشخص کند. بیتهای معتبر باید قسمتی از UIDCLn دریافتی قبل از وقوع تداخل به اضافه یک b (صفر) یا b (۱) باشد، که توسط PCD انتخاب می شود. معمولاً در عمل b (۱) اضافه می کنند.

گام ۸ : PCD NVB و SEL به همراه خود بیتهای معتبر ارسال می کند.

گام ۹ : فقط PICC هایی که قسمت UID CLn آنها برابر با سیستمهای معتبر ارسالی توسط PCD هستند باید بقیه ی UIDCLn خود را ارسال نمایند.

گام ۱۰ : اگر باز تداخل رخ داد، مراحل ۶ تا ۹ تکرار شود. حداکثر تعداد حلقه ها ۳۲ می‌باشد.

گام ۱۱ : اگر تداخلی دیگر رخ نداد، NVB PCD را با مقدار «۷۰» معین می کند.

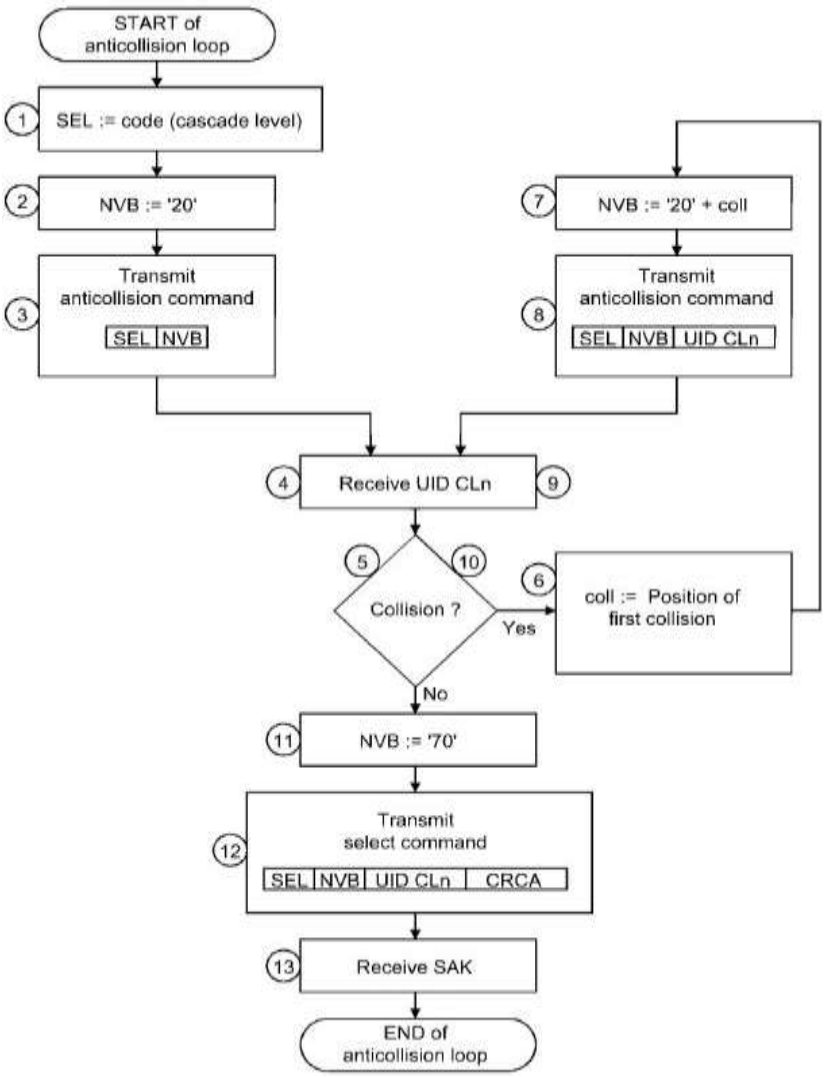
تذکر : این مقدار تعیین می کند که UID CLn PCD کامل را ارسال خواهد کرد.

گام ۱۲ : NVB , SEL PCD به همراه ۴۰ بیت UID CLn و همچنین CRC-A کنترلی ارسال می کند.

گام ۱۳ : PICC که UIDCLn آن با ۴۰ بیت مطابقت دارد، SAK خود را ارسال می کند.

گام ۱۴ : اگر UID کامل باشد، PICC باید SAK به همراه بیت Cascade پاک شده ارسال کند و از حالت READY به ACTIVE برود.

گام ۱۵ : PCD باید بررسی کند که آیا با توجه به بیت Cascade SAK نیازی به حلقه های ضد تداخل در مراحل بعدی Cascade نیاز هست یا خیر.



شکل ۳-۳-۷ :

حلقه ضد تداخل ،

فلوچارت برای PCD

اگر UID یک PICC شناخته شده باشد، PCD می تواند مراحل ۲ تـا ۱۰ را نادیـده بگیـرد و ایـن PICC را بدون انجام حلقه ضد تداخل انتخاب کند.

**۶-۴-۳-۲- رمزنگاری SEL (کد انتخاب)**

طول : ۱ بایت مقدارهای ممکن : b7,b5,b3

| b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | Meaning |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | '93': Select cascade level 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | '95': Select cascade level 2 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | '97': Select cascade level 3 |
| 1 | 0 | 0 | 1 | other values | | | | RFU |

جدول ۳-۳- ۶ : کدگذاری SEL

### ۶-۴-۳-۳- رمزنگاری NVB (تعداد بیتهای معتبر)

طول : ۱ بایت

۴ بایت بالایی عد بایت نامیده می شوند و تعداد همه ی بیتهای داده معتبر تقسیم بر ۸ به اضافه ی NVB,SEL ارسال شده توسط PCD را مشخص می کند. در نتیجه، حداقل مقدار عدد بایت ۲ و حداکثر آن ۷ می باشد.

۴ بایت پایینی عدد بیت نامیده می شو ند و تعداد بیتهای معتبر باقی مانده تقسیم بر ۸ که توسط PCD ارسال می شود را مشخص می کند.

| b8 | b7 | b6 | b5 | Meaning |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | bytecount = 2 |
| 0 | 0 | 1 | 1 | bytecount = 3 |
| 0 | 1 | 0 | 0 | bytecount = 4 |
| 0 | 1 | 0 | 1 | bytecount = 5 |
| 0 | 1 | 1 | 0 | bytecount = 6 |
| 0 | 1 | 1 | 1 | bytecount = 7 |

| b4 | b3 | b2 | b1 | Meaning |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | bitcount = 0 |
| 0 | 0 | 0 | 1 | bitcount = 1 |
| 0 | 0 | 1 | 0 | bitcount = 2 |
| 0 | 0 | 1 | 1 | bitcount = 3 |
| 0 | 1 | 0 | 0 | bitcount = 4 |
| 0 | 1 | 0 | 1 | bitcount = 5 |
| 0 | 1 | 1 | 0 | bitcount = 6 |
| 0 | 1 | 1 | 1 | bitcount = 7 |

جدول ۳-۳-۷ : کدگذاری NVB

### ۶-۴-۳-۴- رمزنگاری SAK (تایید انتخاب)

SAK توسط PICC زمانی ارسال می شود که NVB ۴۰ بیت داده معتبر را معین کرده و همه ی این بیتهای داده با UID40 مطابقت داشته باشد. SAK به وسیله فریم های استاندارد و به همراه CRC-A ارسال می شود.

PCD باید بیت b3 را تشخیص کامل بودن UID بررسی کند. رمزنگاری بیتهای b6,b3 در جدول زیر داده شده است.

| b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | Meaning |
|---|---|---|---|---|---|---|---|---|
| x | x | x | x | x | 1 | x | x | Cascade bit set: UID not complete |
| x | x | 1 | x | x | 0 | x | x | UID complete,<br>PICC compliant with ISO/IEC 14443-4 |

جدول ۳-۳-۸ : کدگذاری SAK

اگر UID کامل نباشد، PICC باید در حالت READY باقی مانده و PCD باید یک حلقه جدید ضد تداخل با افزایش مرحله Cascade به راه اندازد.

### ۶-۴-۳-۵- محتویات UID و مرحله Cascade

UID شامل 10,7,4 بابت UID می باشد. در نتیجه، PICC باید حداکثر تا ۳ مرحله Cascade بگذراند تا همه بایتهای UID آن بدست آید. در هر مرحله آبشتر، یک قسمت از UID شامل ۵ بایت باید به PCD ارسال شود. مطابق حداکثر مرحله Cascade، سه اندازه UID تعریف می شود. این اندازه UID باید عدد ثابت باشد.

| Maximum cascade level | UID size | Number of UID bytes |
|---|---|---|
| 1 | single | 4 |
| 2 | double | 7 |
| 3 | triple | 10 |

جدول ۳-۳-۹ : اندازه UID

برای محتویات UID تعاریف زیر انجام می گیرد :

$1 \le n \le 3$ قسمتی از UID طبق مرحله n Cascade، شامل ۵ بایت، UID CLn

بایت n ام UID، $VID_n \quad n \ge 0$

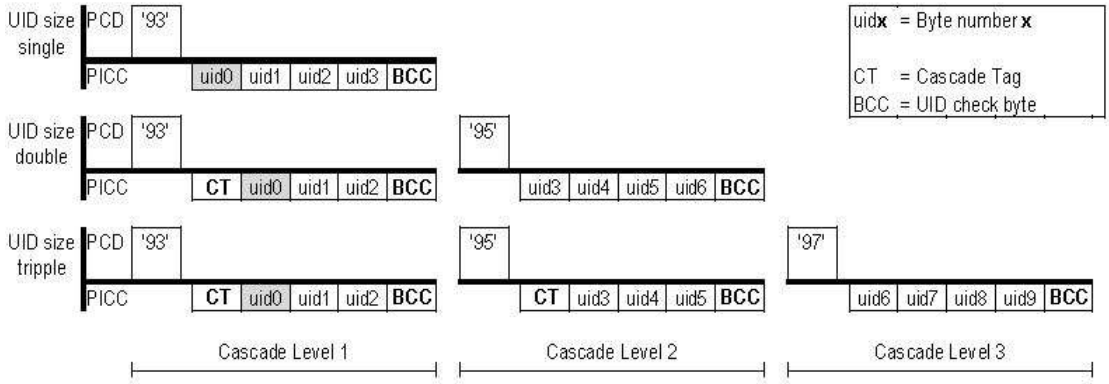
بایت بررسی UID CLn، محاسبه شده به صورت انحصاری یا با ۴ بایت قبلی

برچسب Cascade، ۸۸ CT

UID یک عدد ثابت منحصر به فرد یا یک عدد اتفاقی است که توسط PICC تولید می‌شود. اولین بایت UID (uid) نشان دهنده محتویات بایتهای بعدی UID می باشد.

مقدار ۸۸ برچسب CT Cascade نباید برای $uid\theta$ در اندازه تکی UID استفاده شود.

مقادیر ۸۱ تا «FE» در استفاده های اختصاصی به کار می روند.




شکل ۳-۳-۸ : استفاده از مراحل Cascade

تذکر : هدف از برچسب Cascade تحمیل یک تداخل برای PICC هایی که اندازه کمتری دارند، می باشد. در نتیجه، UID 0 یا UID3 نباید مقدار برچسب Cascade را داشته باشند.

الگوریتم زیر برای بدست آوردن UID کامل در PCD باید اعمال شود :

گام ۱ : PCD مرحله Cascade ۱ را انتخاب می کند.

گام ۲ : حلقه ضد تداخل باید انجام گیرد.

گام ۳ : PCD باید بیت SAK Cascade را بررسی کند.

گام ۴ : اگر بیت Cascade یک بود، PCD باید مرحله Cascade را افزایش داده و یک حلقه ی ضد تداخل جدید ایجاد کند.

گام ۵ : زمانی که یک PICC به وسیله UID کاملش انتخاب شد، باید SAK به همراه بیت Cascade صفر را ارسال کند و از حالت READY به حالت ACTIVE برود.

# استاندارد 14443-4

## کارت های شناسایی – مدارات مجتمع کارت های بدون تماس – کارت های نزدیک

## قسمت چهارم : پروتکل انتقال

### ۱ – هدف

در این قسمت از ISO/IEC14443 مسائل زیر توضیح داده مي شود :

- انواع روش های فعال کردن پروتکل PICC نوع A.
- نحوه تشخیص خطا در این پروتکل و بازیابی آن
- محتویات قالب های RATS و ATS
- بایتهاي ارتباطي TA(1)، TB(1)، TC(1)
- دیگر پارامترهاي مورد نیاز براي برقراري ارتباط بین PICC,PCD
- غیر فعال سازی پروتکل PICC نوع A

۲ – استاندارد های زیر شامل قوانینی می شوند که در این متن ، قوانین این قسمت از استاندارد ISO/IEC 14443 را تشکیل می دهد .

ISO/IEC 14443 کارت های شناسایی – کارتهای بدون تماس مدار مجتمع – کارتهای نزدیک

ISO/IEC ۱۰۳۷۳ کارت های شناسایی – روش های آزمون

ISO/IEC 7816-2 کارت های شناسایی – کارت های مجتمع شده با تماس

### ۳- تعاریف و عبارات

#### ۳-۱- طول بیت

طول بيت به عنوان يك واحد زماني اصلي شناخته مي شود (etu) etu از طريق فرمول زير محاسبه مي شود :

1 etu =128/(D=fc)

مقدار ابتدايي D بايد ۱ باشد. بنابراين etu ابتدايي عبارتست از :

1 etu = 128/fc

فركانس كرير fc در ISO/IEC 14443 مشخص شده است.

### ۳-۲- بلوك

يك نوع فريم ويژه مي باشد كه شاقمل يك قالب پروتكل داده معتبر است يك قالب پروتكل داده معتبر شامل بلوك I، بلكو R و بلوك S مي باشد.

### ۳-۳- بلوك نامعتبر

يك نوع فريم كه شامل قالب پروتكل نامعتبر است. يك ايست، زماني كه هيچ فرمي دريافت نشده باشد، به معني بلوك نامعتبر نمي باشد.

### ۳-۴- فريم

در ISO/IEC14443-3 توضيح داده شده است PICC نوع A از فريم استاندارد تعريف شده براي نوع A استفاده مي كند.

## ۴ – نماد ها

| | |
|---|---|
| ACK | positive ACKnowledgement |
| ATS | Answer To Select |
| CID | Card IDentifier |
| D | Divisor |
| etu | elementary time unit |

| | |
|---|---|
| fc | carrier frequency |
| FSC | Frame Size for proximity Card |
| FSCI | Frame Size for proximity Card Integer |
| FSD | Frame Size for proximity coupling Device |
| FSDI | Frame Size for proximity coupling Device Integer |
| FWI | Frame Waiting time Integer |
| FWT | Frame Waiting Time |
| $FWT_{temp}$ | temporary Frame Waiting Time |
| I-block | Information block |
| INF | INformation Field |
| NAD | Node Address |
| PCD | Proximity Coupling Device |
| PICC | Proximity Card |
| PPS | Protocol and Parameter Selection |
| PPSS | Protocol and Parameter Selection Start |
| PPS0 | Protocol and Parameter Selection 0 |
| PPS1 | Protocol and Parameter Selection 1 |
| R-block | Receive ready block |
| RATS | Request for Answer To Select |
| RFU | Reserved for Future Use |
| S-block | Supervisory block |
| SAK | Select AcKnowledge |
| SFGI | Start-up Frame Guard time Integer |
| SFGT | Start-up Frame Guard Time |

**۵ فعال شدن پروتکل PICC نوع A**

مراحل فعال سازی زیر باید اعمال شود :

* مراحل فعال سازی توضیح داده شده در ISO/IEC14443-3 (در خواست حلقه ضد تداخل و انتخاب)

* در شروع بایت SAK وجود یک ATS باید بررسی شود. SAK در ISO/IEC14443-3 توضیح داده شده است.

* PICC ممکن است به حالت HALT رفته باشند، با استفاده از دستور HALT که در بخش ۳ توضیح داده شده است، اگر هیچ ATS موجود نباشد.

* RATS ممکن است توسط PCD به عنوان دستور بعدی بعد از دریافت SAK که ATS در آن موجود است، ارسال شود.

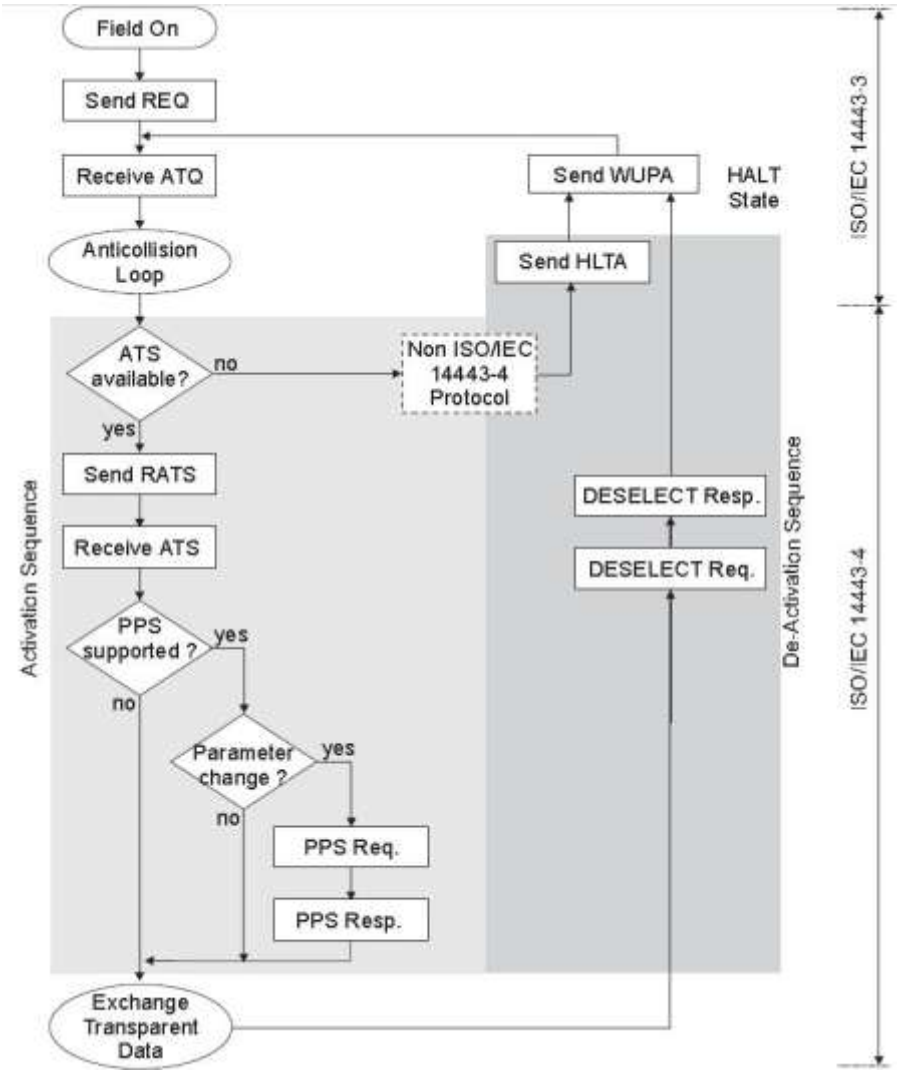
* PICC باید ATS خود را به عنوان پاسخ RATS ارسال کند. PICC باید فقط به RATS پاسخ دهد اگر RATS دقیقاً بعد از انتخاب دریافت شود.

* اگر PICC از پارامترهای قابل تغییر در AIS پشتیبانی می کند، یک درخواست PPS ممکن است توسط PCD به عنوان دستور بعدی بعد از دریافت ATS برای تغییر پارامترها استفاده شود.

* PICC باید یک پاسخ PPS به عنوان جواب PPS در خواستی ارسال کند.

یک PICC نیازی به انجام عملیات PPS ندارد زمانی که پارامترهای قابل تغییر در ATS را پشتیبانی نمی کند.

شکل ۳-۴-۱ : فعالسازی PICC نوع A توسط PCD




**۵-۱- در خواست برای پاسخ به انتخاب (RATS)**

این بند RATS با تمام جزئیات را بیان می کند.

بایت پارامتر شامل دو قسمت می شود :

* نیم بایت بالا b5 تا b8، FSDI نامیده می شود و FSD را رمز نگاری می کند. FSD بزرگترین اندازه فریم می باشد که PCD می تواند دریافت کند. رمزنگاری FSD در جدول ۱ داد شده است .

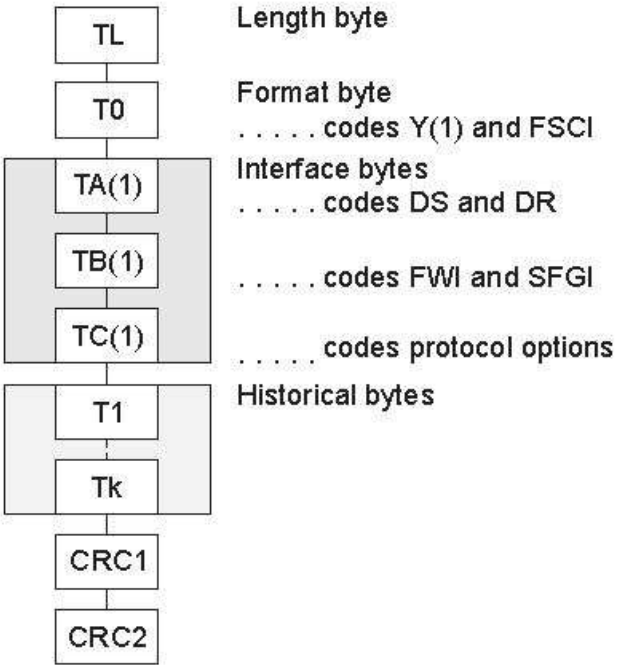
* نیم بایت پایینی b4 تا b1 ، CID نامیده می شود و نشان دهنده عدد منطقی PICC مورد نظر در محدوده ۰ تا ۱۴ می باشد. مقدار ۱۵ RFU می باشد. CID توسط PCD مشخص می شود و باید برای همه

ی PICC ها منحصر به فرد باشد که به طور همزمان در حالت ACTIVE هستند. CID برای زمانی که PICC فعال است ثابت می باشد و PICC باید از CID به عنوان شناساگر خود استفاده کند که در اولین RATS بدون خطا دریافت می شود.

### ۵-۲- پاسخ به ا نتخاب ATS

این بند ATS با همه جوانبش را مشخص می کند.

در حالتی که یکی از موارد در یک ATS ارسالی توسط یک PICC موجود نباشد، مقدار پیش فرض باید برای آن مورد اعمال شود.




شکل ۳-۴-۲ : بدنه ی ATS

### ۵-۲-۱- بدنه ی بایت ها

اندازه طول بایت TL به همراه تعداد متعددی بایتهای اختیاری بعدی به ترتیب زیر می باشد:

* قالب بایت To

* بایتهای ارتباطی TA(1)، TB(1)، TC(1)

* بایتهای گذشته T1 تا Tk

**۵-۲-۲- طول بایت**

طول بایت TL اجباری است و طول ATS ارسالی به همراه خودش را مشخص می کند. دو بایت CRC به TL اضافه نمی شوند.

حداکثر اندازه ATS نباید از FSD نشان داده شده تجاوز کند. بنابراین حداکثر مقدار TL نباید از FSD-2 تجاوز کند.

**۵-۲-۳- قالب بایت**

قالب بایت To اختیاری است و از زمانی که طول بیشتر از یک شود، موجود می باشد. ATS فقط می تواند بایتهای اختیاری زیر باشد، در زمانی که این قالب بایت موجود باشد.

To شامل سه قسمت می باشد :

* با ارزش ترین بیت b8 باید صفر باشد. مقدار یک REU می باشد.

* بیتهای b5,b6,b7 نشان دهنده ی حضور بایت های ارتباطی بعد TA(1)، TB(1) ، TC (1) می باشد.

* نیم بایت کم ارزش پینی b4 تا b1، FSCI نامیده می شود و FSC را رمز نگاری می‌کند، FSC حداکثر اندازه یک فریم قبول شده توسط PICC می باشد. مقدار پیش فرض FSCI، ۲ می باشد که نشان دهنده ی یک FSC ۳۲ بایتی می باشد. رمزنگاری FSC مشابه رمزنگاری FSD می باشد.

**۵-۲-۴- بایت ارتباطی TA(1)**

این بایت شامل ۴ قسمت می شود.

* بیت با ارزش b8، احتمال به کارگیری مقسوم علیه های مختلف را در هر جهت رمزنگاری می کند. وقتی این بیت یکی می شود PICC قادر به این کار نمی باشد .

* بیت های b5 تا b7 ظرفیت نرخ بیت PICC برای جهت PICC به PCD رمزنگاری می‌کند که DS نامیده می شود. مقدار پیش فرض باید (۰) باشد.

* بیت b4 باید صفر شود و مقدار دیگر برای RFU می باشد.

* بیتهای b1 تا b3 ظرفیت نرخ بیت PCD به PICC را که DR نام دارد، کدگذاری می‌کند. مقدار پیش فرض باید b (۰) باشد.

انتخاب یک مخرج خاص D در هر جهتی توسط PCD از طریق PPS امکان پذیر است.

**۵-۲-۵ بایت ارتباطی TB(1)**

این بایت برای مشخص کردن زمان تاخیر فریم و زمان گارد فریم آغازین می باشد.

بایت ارتباطی TB(1) شامل دو قسمت می باشد :

* نیم بایت بالایی b5 تا b8، FWI نامیده می شود و FWT را رمزنگاری می کند.

* نیم بایت پایینی b1 تا b4، SFGI نامیده می شود و یک متغیر ضرب کننده برای تعیین SFGT را رمزنگاری می کند. SFGT زمان گارد خاص مورد نیاز PICC قبل از آماده شدن برای دریافت فریم بعدی بعد از ارسال ATS را مشخص می کند. SFGI در محدوده ۰ تا ۱۴ رمزنگاری می شود. مقدار ۱۵ برای RFU می باشد. مقدار صفر نشان دهنده این است که هیچ SFGT مورد نیاز نیست و مقادیر در محدوده ۱ تا ۱۴ برای محاسبه SFGT استفاده می شوند. مقدار پیش فرض SFGI برابر صفر است.

SFGI از طریق فرمول های زیر محاسبه می شود :

$$SFGI = (256*16/fc)*2^{SFGI}$$

$$SFGT_{MAX} =\sim 4g4gms$$

**۵-۲-۶- بایت ارتباطی TC(1)**

این بایت یک پارامتر پروتکل را مشخص می کند.

بایت ارتباطی TC(1) خاص شامل دو بخش می شود :

* بایت های با ارزش b8 تا b3، b (000000) هستند و دیگر مقادیر REU می باشند.

* بیتهای b1,b2 نشان دهنده زمینه اختیاری که PICC از آن پشتیبانی می کند، می‌باشد. PCD نباید زمینه ای که PICC از آن پشتیبانی نمی کند را ارسال کند. مقدار پیش فرض باید b(10) باشد که نشان دهنده پشتیبانی از CID و پشتیبانی نکردن از NAD می‌باشد.

**۵-۳- زمان انتظار فریم فعال سازی**

این زمان مشخص کننده ی حداکثر زمان برای یک PICC برای شروع ارسال پاسخش بعد از پایان یک فریم رسیده توسط PCD می باشد و مقدار fc / ۶۵۵۳۶ ( $4833\mu s$ ~) را دارد.

تذکر : حداقل زمان بین فریم ها در هر جهت در ISO/IEC14443-3 مشخص شده است.

**۵-۴- شناسایی و بازیابی خطا**

**۵-۴-۱- کار با ATS, RATS**

**۵-۴-۱-۱- قوانین PCP**

وقتی PCP، PATS را فرستاد و یک ATS معتبر دریافت کرد، PCD باید عملیات را ادامه دهد.

در موارد دیگر PCD ممکن است یک در خواست PPS را درباره ارسال کند و به عملیات ادامه دهد.

**۵-۴-۱-۲- قوانین PICC**

وقتی که PICC یک RATS دریافت کرد، ATS خود را ارسال کرده و

۱- یک در خواست PPS معتبر دریافت می کند، PICC باید :

* پاسخ PPS را ارسال کند.

* درخواست PPS را غیر فعال کند (پاسخ طولانی به درخواست های دریافتی PPS ندهد) و

* پارامتر دریافتی را فعال کند.

۲- یک بلوک نامعتبر دریافت کند، PICC باید.

* در خواست PPS را غیر فعال کند و

* در حالت دریافت باقی بماند.

۳- یک بلوک معتبر دریافت کرده، به جز یک در خواست PPS، PICC باید :

* درخواست PPS را غیر فعال کند و

* به علمیات ادامه دهد.

**۵-۴-۳- کار با CID در مدت فعالسازی**

وقتی PCD یک RATS ارسال کند که شامل یک CID=n که با صفر برابر نیست و :

۱- یک AIS دریافت کند که نشان دهد CID پشتیبانی می شود. PCD باید

* بلوکهایی که شامل CID=n است را به این PICC ارسال کند و

از CID=n برای RATS های بعدی در زمانی که PICC در حالت ACTIVE هست استفاده نکند.

۲- یک ATS دریافت کند که نشان دهد از CID پشتیبانی نمی کند، PCD باید :

* بلوکهایی شامل هیچ CID نیست به این PICC ارسال کند و تا زمانی که این PICC در حالت ACTIVE است، هیچ PICC دیگری را فعال نکند.

وقتی PCD یک RATS که شامل یک CID برابر صفر ارسال کند و

۱- یک ATS دریافت کند که نشان دهد CID پشتیبانی می شود، PCD

* بلوکهایی شامل CID برابر با صفر به این PICC ارسال کند و

* تا زمانی که این PICC در حالت ACTIVE است، هیچ PICC دیگری را فعال نکند.

۲- یک ATS دریافت کند که نشان دهد CID پشتیبانی نمی شود، PCD باید

* بلوکهایی که شامل هیچ CID شود به این PICC ارسال کند.

**۶- غیر فعال کردن پروتکل نوع PICC A**

PICC باید به حالت HALT رفته بعد از اینکه معاملات بین PCD و PICC کامل شده باشد.

غیر فعال کردن یک PICC با استفاده از دستور DESELECT انجام می پذیرد.

دستور DESELCT به عنوان یک پروتکل بلوک S رمز نگاری می شود و شامل یک بلوک در خواست S (DESELECT) ارسالی توسط PCD و یک پاسخ S(DESELCT) به عنوان تایید ارسالی توسط PICC می باشد.

**۶-۱- زمان انتظار فرمی غیر فعال ساز**

این سازمان حداکثر زمان برای یک PICC برای آغاز ارسال فریم پاسخ S(DESELECT) خودش بعد از اتمام فریم در خوایت S(DESELECT) در خواستی از PCD می باشد و مقدار 65536/fc ($\sim 4833\mu s$) را دارد.

**۶-۲- بازیابی و شناسایی خطا**

وقتی PCD یک در خواست (DESELCT) S ارسال می کند و یـک پاسـخ (DESELECT) S دریافـت کنـد، PICC به طور موفق به حالت HALT وارد می شود و CID مربوط به آن آزاد می شود.

وقتی PCD موفق به دریافت یک پاسخ S(DESELCT) نشود، PCD ممکن است مراحل غیر فعال سـازی را دوباره تکرار کند.

# فصل چهارم

# نمونه عملی RFID ضدتداخل

**۱-۴) مقدمه**

در این فصل به بررسی یک نمونه مدار عملی در زمینه سیستم های RFID ضد تداخل می پردازیم . این مدار را می توان به دو بخش تقسیم کرد : بخش اول شامل قسمت پردازشی و کنترلی سیستم می شود و در بخش دوم شامل یک سیستم RFID که قابلیت ضدتداخل داشته باشد می باشد . در بخش پردازشی مدار من از یک میکروکنترلر ATMega32 که بسیار در بازار رایج است استفاده کرده ام . همچنین این بخش شامل یک LCD برای نمایش اطلاعات درون یک تگ که در اینجا به صورت کارت می باشد ، استفاده می شود . در بخش سیستم RFID من از ماژول RFID ، YLMF018 استفاده کرده ام . همچنین از تگ کارت های s50 دارای حافظه ی ۱ کیلو بیاتی استفاده شده است . همچنین ۳ عدد کلید نیز روی پایه های میکرو قرار دارد .

**۲-۴ نحوه کار مدار**

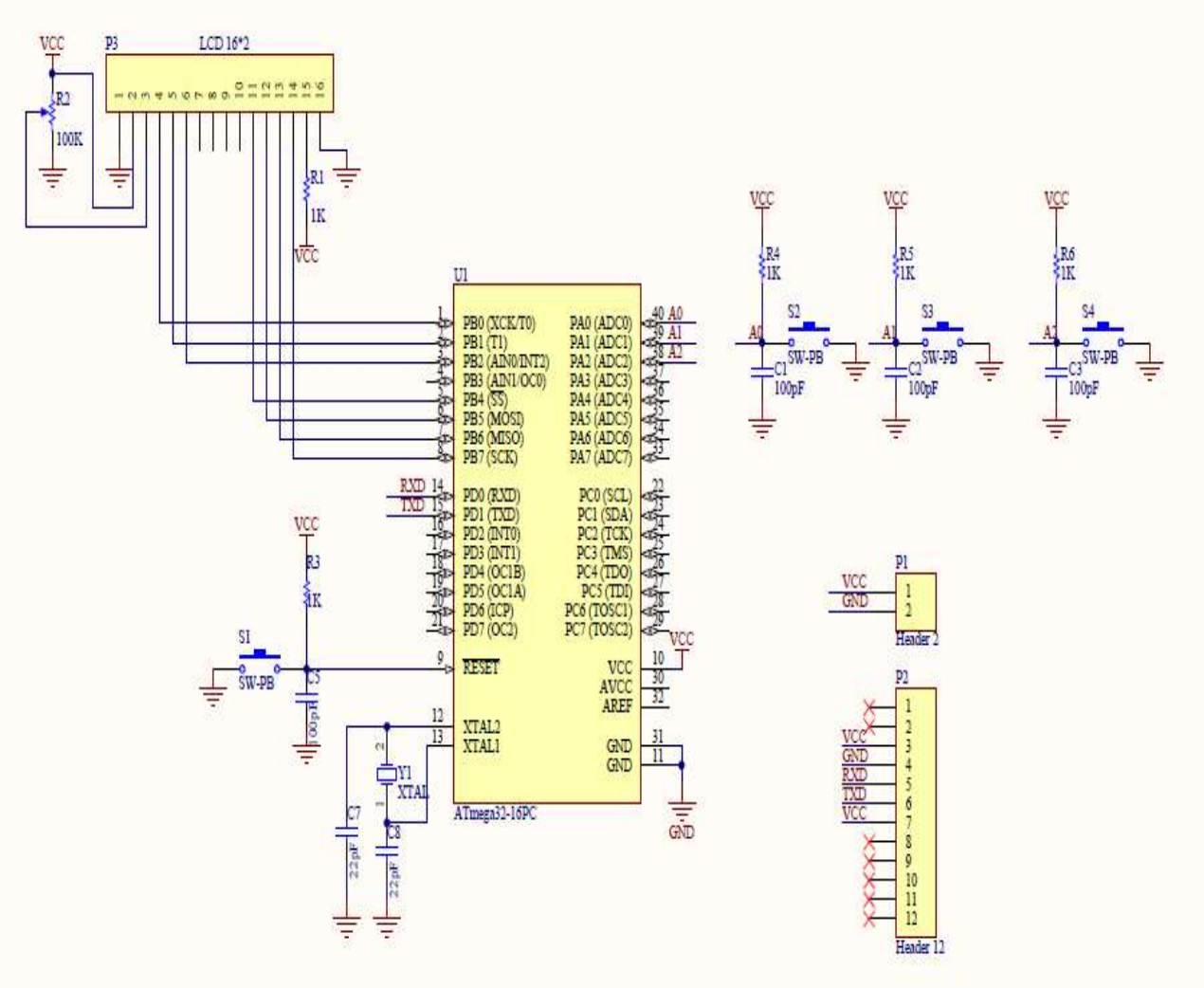
نحوه کار مدار به این صورت است که در ابتدای کار پس از راه افتادن مدار و قبل از این که هیچ کارتی را روی دستگاه قرار ندادیم ، مدار عدم حضور کارت را به ما اطلاع دهد . حال اگر ما کارتی را در محدوده عملیاتی این مدار که حدود ۱۰ سانتی متر می باشد وارد کنیم ، سیستم به کارت مورد نظر متصل شده و مقدار عددی که قبلا درون این کارت ذخیره شده را بر روی LCD نشان می دهد. در این حالت ما سه کار مختلف با سه کلید تعبیه شده بر روی پایه های میکرو می توانیم انجام دهیم. با زدن کلید اول مقدار درون حافظه کارت با فشار دادن هر بار کلید یک واحد از عدد درون کارت کم شود . با فشار دادن کلید دوم مثلا به عنوان شارژ کارت ۱۰۰ واحد درون این کارت واریز شود ، و اگر از قبل مقداری درون این کارت باشد این عدد به آن اضافه شود . و در آخر با فشار دادن کلید سوم عدد درون کارت به طور کل پاک شود و عدد صفر جایگزین آن شود . همه ی این کار ها را باید بتوان در حضور دیگر کارت ها نیز انجام داد یعنی اگر سیستم به یک کارت متصل است حضور دیگر کارت های درون میدان باعث تداخل در ارتباط بین کارت و سیستم نشوند . و اینگونه سیستم ضدتداخل بودن خود را اثبات خواهد کرد .

## ۴-۳ بررسی اجزا مدار پردازشگر

همانطور که ذکر شد از یک میکروکنترلر Atmega32 به عنوان هسته مرکزی قسمت پردازشی مدار استفاده شده است . برای راه اندازی این میکروکنترلر از یک کریستال ۸ مگاهرتز استفاده شده تا دقت کار مدار را به بالاترین حد ممکن برساند . از کلید های Push Bottun (کلید فشاری) Normally open بر روی پایه های مدار استفاده شده و برای کارایی بهتر ای پایه ها را Pull Up خارجی کردم. از یک LCD کاراکتری 16*2 به عنوان نمایشگر سیستم استفاده کردم .

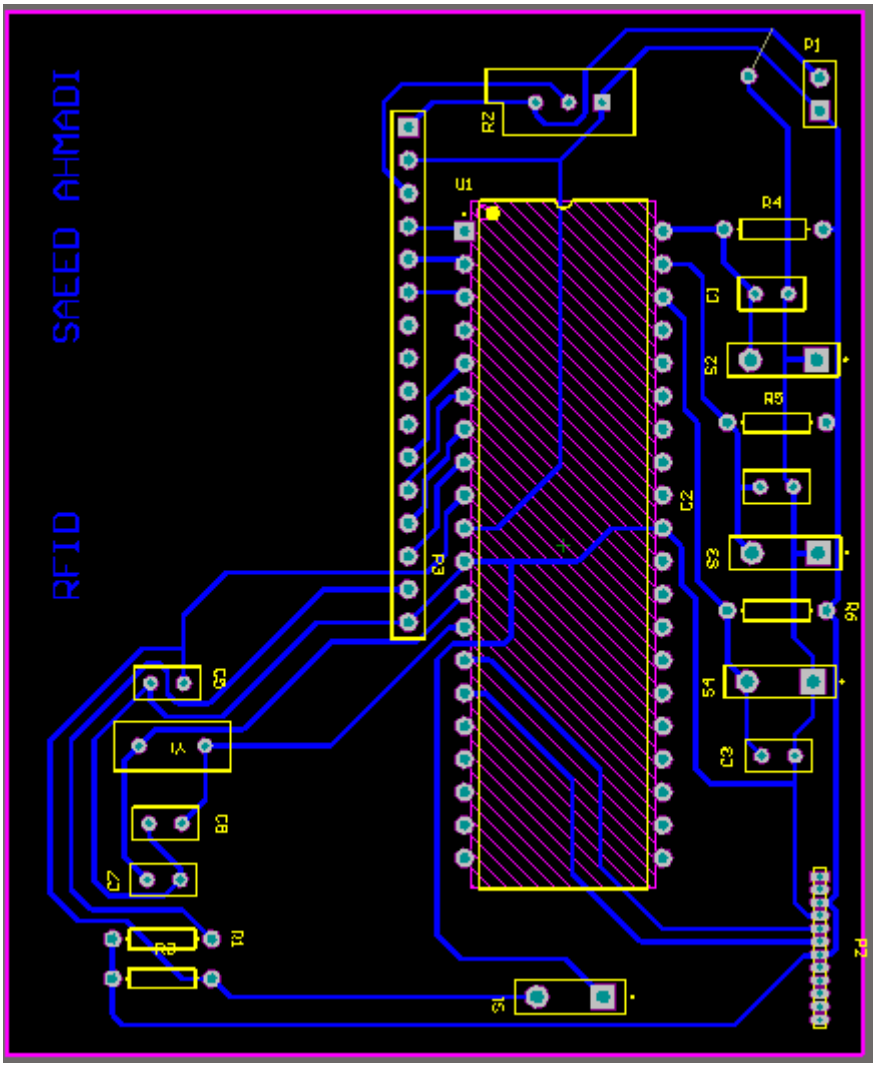
## ۴-۴ بررسی نمای شماتیک و PCB مدار پردازشگر

در تصویر زیر شماتیک مدار را می بینید . همانطور که مشاهده می فرمایید Lcd به پورت B متصل شده است و همچنین سه کلید فشاری ما به ترتیب به پایه های PA0 ، PA1 و PA2 متصل اند.




شکل ۴-۱ : نمای شماتیک مدار پردازشگر

در صفحه بعد نیز نمای PCB مدار را خواهید دید . لازم به ذکر است که این کار توسط برنامه Altium Designer انجام شده است .




شکل ۴-۲ : نمای PCB مدار پردازشگر

به این دلیل مدار را بزرگ و با جاهای خالی زیاد طراحی کردم که هم LCD خارج از برد نیفتد و همچنین از لحاظ هزینه هیچ تفاوتی با یک برد کوچکتر نمی کرد .

### ۴-۵ ماژول YLMF018

همانطور که در بالا ذکر شد از ماژول YLMF018 در بخش RFID استفاده شده است . این ماژول از آی سی MFRC500 به عنوان هسته ی مرکزی سیستم RFID خود بهره می برد . این آی سی ساخت شرکت NXP می باشد و از تمام بخش های استاندارد ISO/IEC 14443 A پشتیبانی می کند . این

آی سی یا به طور کلی این استاندارد با فرکانس 13.56 مگاهرتز کار می کند . این ماژول از استاندارد Mifare که در ادامه به ویژگی های آن می پردازیم ، پشتیبانی می کند .

### ۴-۵-۱ استاندارد Mifare

- ۱۰۲۴ کیلو بایت حافظه EEPROM که به ۱۶ قسمت ۶۴ کیلوبایتی تقسیم شده است
- قابلیت نوشتن ۱۰۰۰۰۰ بار
- قابلیت نگهداری اطلاعات تا ۱۰ سال
- پشتیبانی از ISO/IEC 14443 A
- نرخ داده ۱۰۶ کیلوبیتی
- ضد تداخل بیتی
- محدوده عملیاتی تا حداکثر ۱۰ سانتی متر
- شماره سریال منحصر به فرد ۴ بایتی
- مولد عدد تصادفی
- کلید دسترسی ۲ بایتی برای هر بخش
- حالت دسترسی انفرادی به هر بخش

### ۴-۵-۲ ویژگی های فنی

- تغذیه : ۵ ولتی ، ۸۰ تا ۱۰۰ میلی آمپر
- رابط RS232 یا TTL232
- سرعت انتقال : 19200 bps
- فاصله خواندن و نوشتن تا حداکثر ۱۰۰ میلی متر بسته به تگ

- دمای ذخیره : 80+ ~ 40- درجه سانتی گراد
- دمای کار : 70+~0 درجه سانتی گراد

### ۴-۵-۳ تنظیمات ارتباط

پروتکل ارتباطی بیات گرا می باشد . هر دو بایت های ارسالی و دریافتی به صورت هگز می باشند . پارامتر های ارتباط به ضورت زیر می باشند :

Baud rate: 19200 bps

Data: 8 bits

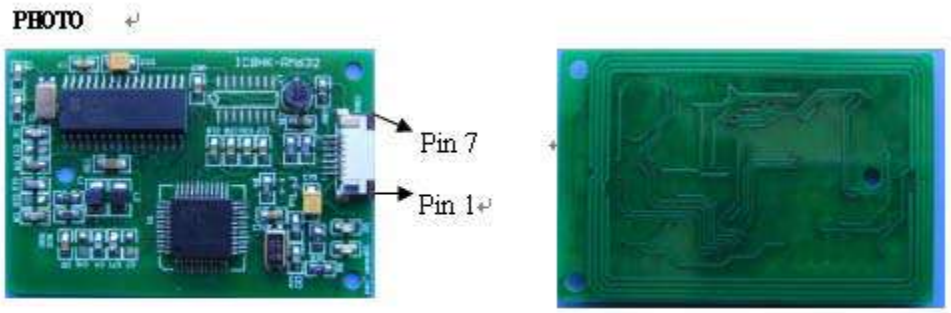
Stop: 1 bit

Parity: None

Flow control: None

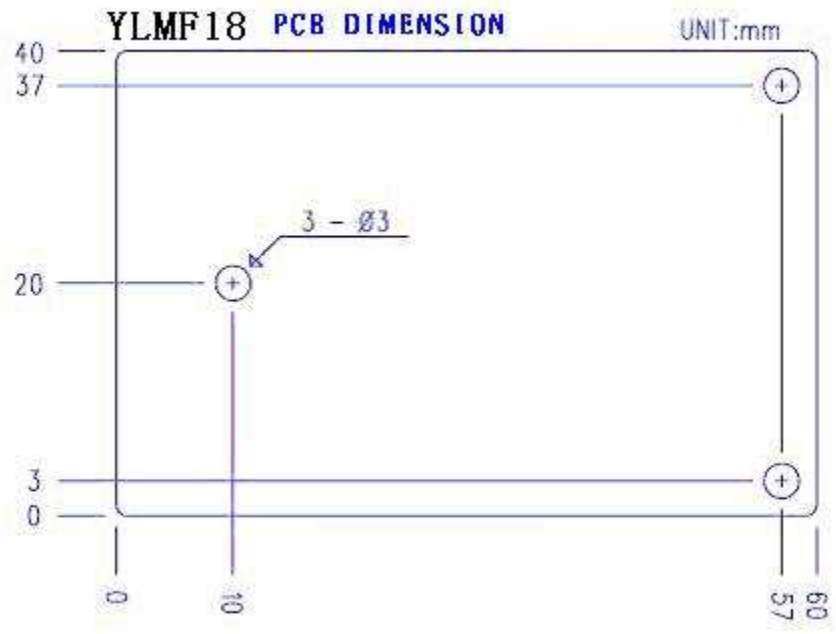
### ۴-۵-۴ تصویر و ابعاد ماژول

در زیر تصویر یک ماژول YLMF018 را مشاهده می کنید .




شکل ۴-۳ : ماژول YLMF018

ابعاد این ماژول به صورت زیر می باشد :




شکل ۴-۴ : ابعاد ماژول YLMF018

در جدول زیر طریقه پین های خروجی را نشان می دهد :

| NO | Define |
|---|---|
| 1 | Reserve |
| 2 | Reserve |
| 3 | VCC(+5V) |
| 4 | GND |
| 5 | RX |
| 6 | TX |
| 7 | VCC(+5V) |

جدول ۴-۱ : پایه های خروجی YLMF018

### ۴-۵-۵ ویژگی ماژول

| Parameter | Min | Type | Max | Units |
|---|---|---|---|---|
| Voltage | 4. 5 | 5. 0 | 5. 5 | V |
| Current (include antenna) | | 90 | | mA |
| Initialization time | 100 | | 500 | MS |
| Opertating temperature | −25 | | +85 | ℃ |
| Storage temperature | −40 | | +125 | ℃ |

جدول ۴-۲ : ویژگی YLMF018

### ۴-۵-۶ پروتکل ارتباطی

هر بایت ارسالی یا دریافتی به صورت های زیر می باشند :

**COMMAND :**

| | Data length (Byte) | | X O R | S U M |
|---|---|---|---|---|
| **Head** | 02 | Fixed: 0xAA , 0xBB | | |
| **Length** | 02 | There are several effective bytes that including XOR follows this column. | FF | 00 |
| **Node ID** | 02 | Destination Node Address Number.<br>xx xx: Low byte first<br>00 00: Broadcast to each reader. | X | S |
| **Function code** | 02 | It will be transmission ability of each different command . Low byte frist | X | S |
| **Data** | 00~D0 | Data length is not fixed, according to its purpose. | X | S |
| **XOR** | 01 | XOR each byte from Node ID to Last Data byte with 0xFF. | | S |
| | | | | |

جدول ۴-۳ : قالب دستور ارسالی

REPLY DATA FORMAT :

| | Data length (Byte) | | | |
|---|---|---|---|---|
| **Head** | 02 | Fixed: 0xAA , 0xBB | | |
| **Length** | 02 | There are several effective bytes that including XOR follows this column. | | |
| **Node ID** | 02 | Destination Node Address Number.<br>xx xx: Low byte first<br>00 00: Broadcast to each reader. | | |
| **Function code** | 02 | It will be transmission ability of each different command . Low byte frist | | |
| **Status** | 1 | Reply result , if succeed is 0 ,other fail . | | |
| **Data** | 00~D0 | Data length is not fixed, according to its purpose. | | |
| **XOR** | 01 | XOR each byte from Node ID to Last Data byte | | |
| | | | | |

جدول ۴-۴ : قالب اطلاعات دریافتی

### ۴-۵-۷ دستورات

**دستور راه اندازی پورت ارتباطی 0x0101 :**

عملکرد : تنظیم Baud Rate

قالب : aa bb 06 00 00 01 01 “Baud_para” “xor Chk”

Baud_parameter :
0 = 4800;
1 = 9600;
2 = 14400;
3 = 19200;
4 = 28800;
5 = 38400;
6 = 57600;
7 = 115200;

**دستور خواندن نوع دستگاه 0x0104 :**

عملکرد : مدل و ورژن دستگاه را می خواند .

**دستور تنظیم حالت آنتن 0x010c :**

عملکرد : ON و OFF کردن آنتن

قالب : aa bb 06 00 00 00 0c 01 x 0D

اگر x صفر باشد آنتن در وضعیت خاموش و اگر یک باشد آنتن روشن خواهد شد .

**دستور درخواست MIFARE 0x0201 :**

عملکرد : ارسال درخواست کارت های نوع A

قالب : aa bb 06 00 00 00 01 02 req_code XOR

Req_code = 0x52 به همه کارت های درون محدوده درخواست را ارسال می کند .

Req_code = 0x26 فقط به کارت هایی که در حالت idle هستند ارسال می کند .

**دستور ضدتداخل مایفر 0x0202 :**

عملکرد : عملیات استخراج یک کارت از بین دیگر کارت ها

۴ بایت آخر دریافتی ، شماره سریال منحصر به فرد کارت مورد نظر می باشد .

**دستور انتخاب مایفر 0x0203 :**

عملکرد : انتخاب کارت

قالب : aa bb 09 00 00 00 03 02 xx xx xx xx XOR

بایت نهم تا دوازدهم شماره سریال کارت مورد انتخاب باید باشد .

**دستور تصدیق مایفر 0x0207 :**

عملکرد : تصدیق کارت مورد نظر

قالب : aa bb xx 00 00 00 07 02 Auth_mode Block xx xx xx xx xx xx XOR

Auth_mode = حالت تصدیق , کلید A = 0x60 و کلید B = 0x61

Block = بلوک مورد نظرد برای تصدیق

**دستور خواندن مایفر 0x0208 :**

عملکرد : خواندن باوک انتخابی کارت مورد نظر

قالب : aa bb 06 00 00 00 08 02 Block XOR

Block = بلوک مورد نظر برای خواندن

در جواب دریافتی بایت دهم تا بایت بیست و پنجم ، بایت های دیتا می باشند .

**دستور نوشتن مایفر 0x0209 :**

عملکرد : نوشتن در بلوک دلخواه بر روی کارت مورد نظر .

قالب :

aa bb 16 00 00 00 09 02 Block
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 Da Db Dc Dd De Df XOR

Block = بلوک مورد نظر

D0-Df = ۱۶ بایت برای نوشتن می باشند .

**دستور قطع ارتباط مایفر 0x0204 :**

عملکرد : قطع ارتباط بین کارت از قبل وصل شده و ماژول

قالب : aa bb 05 00 00 00 04 02 06

**۶-۴ برنامه میکرو**

برای برنامه نویسی از برنامه CodevisionAVR و مشخصا زبان C استفاده کرده ام . برای اتصال به یک کارت مایفر باید به طریق زیر عمل کرد :

۱. روشن کردن آنتن ماژول
۲. ارسال دستور درخواست مایفر Request mifare
۳. ارسال دستور ضدتداخل مایفر Anticollision mifare

۴. سریال کارت به دست آمده از دستور قبل را در دستور انتخاب مایفر (Select Mifare) استفاده کرده و ارسال کنیم .
۵. انتخاب بلوک دلخواه و ارسال آن با دستور تصدیق مایفر (Authentication Mifare) به همراه کلید انتخابی

بعد از انجام این مراحل می توان در بلوک انتخابی یک کارت اطلاعات نوشت یا اطلاعات آن را خواند .

در برنامه زیر نیز همه این مراحل در یک مرحله ساختگی Connect انجام می شود . این برنامه شامل دو فایل Main.c و Mifare.c می باشد .

Main.c :

```
#include <mega32.h>
#include <delay.h>
#include "mifare.h"
#include <alcd.h>
#include <stdio.h>

// Declare your global variables here
char rx_buffer[26];
unsigned char rx_index;

unsigned char data[16]={48,48,48,48,48,48,48,48,48,48,48,48,48,48,48,48};
unsigned char test[9]={0xaa,0xbb,0x05,0x00,0x00,0x00,0x04,0x01,0x05};
unsigned char temp[20];

void main(void)
```

```
{
// Declare your local variables here
unsigned char i;
bit j=0;

PORTA=0x00;
DDRA=0x00;

PORTB=0x00;
DDRB=0x00;

PORTC=0x00;
DDRC=0x00;

PORTD=0x00;
DDRD=0x00;

TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

TCCR1A=0x00;
```

```
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

MCUCR=0x00;
MCUCSR=0x00;

TIMSK=0x01;
#asm("sei")

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
```

```
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 19200
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x19;

ACSR=0x80;
SFIOR=0x00;

ADCSRA=0x00;

SPCR=0x00;

TWCR=0x00;

lcd_init(16);
lcd_clear();
lcd_gotoxy(0,0);
delay_ms(20);
```

```
lcd_putsf("Be Name Alim");
delay_ms(2000);
lcd_clear();
lcd_gotoxy(0,0);
for(i=0;i<9;i++)
{
putchar(test[i]);
};
for(i=0;i<20;i++)
{
temp[i]=getchar();
};
for(i=9;i<20;i++)
{
lcd_putchar(temp[i]);
delay_ms(5);
};
UCSRB=0x98;
delay_ms(2000);
for(i=0;i<20;i++)
{
temp[i]=0;
};
```

```
if(cmd_setanten(1));
delay_ms(2000);
lcd_clear();
lcd_gotoxy(0,0);
while (1)
{
if(connect(4))
{
lcd_clear();
lcd_gotoxy(0,0);
delay_ms(5);
lcd_putsf("Credit Card :");
delay_ms(1);
lcd_gotoxy(0,1);
cmd_readmifare(4,data);
for(i=16;i>0;i--)
{
if((data[(i-1)]==48)&&j==0) j=0 ;
else
{
j=1;
lcd_putchar(data[(i-1)]);
```

```
}
}

if(j==0) lcd_putsf("0");
if((PINA.0==0)&&j==1)
{
for(i=0;i<16;i++)
{
if(data[i]==48) data[i]=57;
else
{
data[i]-=1;
break;
}
}
cmd_writemifare(4,data);
while(!PINA.0);
}
if(PINA.1==0)
{
for(i=2;i<16;i++)
{
if(data[i]==57) data[i]=48;
```

```
else
{
data[i]+=1;
break;
}
}
cmd_writemifare(4,data);
while(!PINA.1);
}
if(PINA.2==0)
{
for(i=0;i<16;i++)
{
data[i]=48;
}
cmd_writemifare(4,data);
while(!PINA.2);
}
delay_ms(500);
}
else
{
lcd_clear();
```

```
lcd_gotoxy(0,0);
delay_ms(5);
lcd_putsf("No Cards Available!!!");
delay_ms(500);
}
j=0;
}
}
```

: Mifare.c

```
#include "mifare.h"
#include <alcd.h>
#include <delay.h>
#include <stdlib.h>
extern char rx_buffer[];
extern unsigned char rx_index;

unsigned char cont_tim0;unsigned char c_tim0;
bit g_bTimeOut=0;bit g_bReceCommandOk=0;
char cmd_number;

//-----------------------------------------------------------------------------
const  unsigned char baud_table[4]          ={ 0x00 , 0x00 , 0x01 , 0x01 };
```

```
const  unsigned char setled_table[4]          ={ 0x00 , 0x00 , 0x07 , 0x01 };
const  unsigned char setanten_table[4]        ={ 0x00 , 0x00 , 0x0c , 0x01 };
const  unsigned char Request_table[4]         ={ 0x00 , 0x00 , 0x01 , 0x02 };
const  unsigned char anticollision_table[4]   ={ 0x00 , 0x00 , 0x02 , 0x02 };
const  unsigned char select_table[4]          ={ 0x00 , 0x00 , 0x03 , 0x02 };
const  unsigned char Authentication_table[4]  ={ 0x00 , 0x00 , 0x07 , 0x02 };
const  unsigned char readmifare_table[4]      ={ 0x00 , 0x00 , 0x08 , 0x02 };
const  unsigned char writemifare_table[4]     ={ 0x00 , 0x00 , 0x09 , 0x02 };
const  unsigned char halt_table[4]            ={ 0x00 , 0x00 , 0x04 , 0x02 };
//-----------------------------------------------------------------------------------------

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
// Place your code here
TCCR0=0x00;//stop timer0
cont_tim0++;
if (cont_tim0==c_tim0){
g_bTimeOut=1;
cont_tim0=0;
}
else {
g_bTimeOut=0;
```

```
TCCR0=0x03;//start timer0
};
}
//-----------------------------------------------------------------------------

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status=0,data;
status=UCSRA;
data=UDR;
if (!(status & 0x10))
{
rx_buffer[rx_index]=data;
rx_index++;
switch (cmd_number) {
case 0: //cmd_baud
case 1: //cmd_setled
case 2: //cmd_setanten
case 6: //cmd_Authentication
case 8: //cmd_writemifare
case 9: //cmd_halt
if (rx_index==10)
```

```
{
rx_index=0;
if (rx_buffer[8]==0 ) g_bReceCommandOk=1;
cont_tim0=0;TCCR0=0x00;//stop timer0
}
break;
case 3: //cmd_Request
if (rx_index==12)
{
rx_index=0;
if (rx_buffer[8]==0 ) g_bReceCommandOk=1;
cont_tim0=0;TCCR0=0x00;//stop timer0
}
break;
case 4: //cmd_anticollision
if (rx_index==14)
{
rx_index=0;
if (rx_buffer[8]==0 ) g_bReceCommandOk=1;
cont_tim0=0;TCCR0=0x00;//stop timer0
}
break;
case 5: //cmd_select
```

```
if (rx_index==11)
{
rx_index=0;
if (rx_buffer[8]==0 ) g_bReceCommandOk=1;
cont_tim0=0;TCCR0=0x00;//stop timer0
}
break;
case 7: //cmd_readmifare
if (rx_index==26)
{
rx_index=0;
if (rx_buffer[8]==0 ) g_bReceCommandOk=1;
cont_tim0=0;TCCR0=0x00;//stop timer0
}
break;
};
}
}
//****************
char cmd_baud(unsigned char a)
{
char i;
unsigned char xxor=0;
```

```
cmd_number=0;rx_index=0;g_bReceCommandOk=0;
g_bTimeOut=0;cont_tim0=0;c_tim0=100;TCCR0=0x03;//start timer0
putchar(0xaa);
putchar(0xbb);
putchar(0x06);
putchar(0x00);
for(i=0;i<4;i++)
{
putchar(baud_table[i]);
xxor ^= baud_table[i];
}
putchar(a);
xxor ^= a;
putchar(xxor);
while (!g_bReceCommandOk && !g_bTimeOut);
if (!g_bTimeOut)
return true;
else
return false;
}
//****************
char cmd_setled(unsigned char a)
{
```

```
char i;
unsigned char xxor=0;
cmd_number=1;rx_index=0;g_bReceCommandOk=0;
g_bTimeOut=0;cont_tim0=0;c_tim0=100;TCCR0=0x03;//start timer0
putchar(0xaa);
putchar(0xbb);
putchar(0x06);
putchar(0x00);
for(i=0;i<4;i++)
{
putchar(setled_table[i]);
xxor ^= setled_table[i];
}
putchar(a);
xxor ^= a;
putchar(xxor);
while (!g_bReceCommandOk && !g_bTimeOut);
if (!g_bTimeOut)
return true;
else
return false;
}
//****************
```

```
char cmd_setanten(unsigned char a)
{
char i;
unsigned char xxor=0;
cmd_number=2;rx_index=0;g_bReceCommandOk=0;
g_bTimeOut=0;cont_tim0=0;c_tim0=100;TCCR0=0x04;//start timer0
putchar(0xaa);
putchar(0xbb);
putchar(0x06);
putchar(0x00);
for(i=0;i<4;i++)
{
putchar(setanten_table[i]);
xxor ^= setanten_table[i];
}
putchar(a);
xxor ^= a;
putchar(xxor);
while (!g_bReceCommandOk && !g_bTimeOut);
if (!g_bTimeOut)
{
return true;
}
```

```
else
return false;

}
//****************
char cmd_Request(unsigned char a,unsigned int *tag_type)
{
char i;
unsigned char xxor=0;
cmd_number=3;rx_index=0;g_bReceCommandOk=0;
;g_bTimeOut=0;cont_tim0=0;c_tim0=100;TCCR0=0x03;//start timer0
putchar(0xaa);
putchar(0xbb);
putchar(0x06);
putchar(0x00);
for(i=0;i<4;i++)
{
putchar(Request_table[i]);
xxor ^= Request_table[i];
}
putchar(a);
xxor ^= a;
putchar(xxor);
```

```
while (!g_bReceCommandOk && !g_bTimeOut);
if (!g_bTimeOut)
{
*tag_type=(unsigned int)((unsigned int)(rx_buffer[9]<<8)|rx_buffer[10]);
return true;
}
else
return false;

}
//****************
char cmd_anticollision(unsigned char *serial)
{
char i;
unsigned char xxor=0;
cmd_number=4;rx_index=0;g_bReceCommandOk=0;
g_bTimeOut=0;cont_tim0=0;c_tim0=200;TCCR0=0x03;//start timer0
putchar(0xaa);
putchar(0xbb);
putchar(0x05);
putchar(0x00);
for(i=0;i<4;i++)
{
```

```
putchar(anticollision_table[i]);
xxor ^= anticollision_table[i];
}
putchar(xxor);
while (!g_bReceCommandOk && !g_bTimeOut);
if (!g_bTimeOut)
{
*(serial+0)=rx_buffer[9];
*(serial+1)=rx_buffer[10];
*(serial+2)=rx_buffer[11];
*(serial+3)=rx_buffer[12];
return true;
}
else
return false;

}
//****************
char cmd_select(unsigned char a,unsigned char b,unsigned char c,unsigned
char d)
{
char i;
unsigned char xxor=0;
cmd_number=5;rx_index=0;g_bReceCommandOk=0;
```

```
g_bTimeOut=0;cont_tim0=0;c_tim0=100;TCCR0=0x03;//start timer0
putchar(0xaa);
putchar(0xbb);
putchar(0x09);
putchar(0x00);
for(i=0;i<4;i++)
{
putchar(select_table[i]);
xxor ^= select_table[i];
}
putchar(a);xxor ^= a;
putchar(b);xxor ^= b;
putchar(c);xxor ^= c;
putchar(d);xxor ^= d;
putchar(xxor);
while (!g_bReceCommandOk && !g_bTimeOut);
if (!g_bTimeOut)
{
return true;
}
else
return false;
}
```

```
//****************
char cmd_Authentication(unsigned char b_key,unsigned char a_block,unsigned
char a,unsigned char b,unsigned char c,unsigned char d,unsigned char
e,unsigned char f)
{
char i;
unsigned char xxor=0;
cmd_number=6;rx_index=0;g_bReceCommandOk=0;
g_bTimeOut=0;cont_tim0=0;c_tim0=25;TCCR0=0x03;//start timer0
putchar(0xaa);
putchar(0xbb);
putchar(0x0d);
putchar(0x00);
for(i=0;i<4;i++)
{
putchar(Authentication_table[i]);
xxor ^= Authentication_table[i];
}
putchar(b_key);xxor ^= 0x60;
putchar(a_block);xxor ^= a_block;
putchar(a);xxor ^= a;
putchar(b);xxor ^= b;
putchar(c);xxor ^= c;
putchar(d);xxor ^= d;
```

```
putchar(e);xxor ^= e;
putchar(f);xxor ^= f;
putchar(xxor);
while (!g_bReceCommandOk && !g_bTimeOut);
if (!g_bTimeOut)
{
return true;
}
else
return false;
}
//****************
char cmd_readmifare(unsigned char _block, unsigned char *ptr)
{
char i;
unsigned char xxor=0;
cmd_number=7;rx_index=0;g_bReceCommandOk=0;
g_bTimeOut=0;cont_tim0=0;c_tim0=100;TCCR0=0x03;//start timer0
putchar(0xaa);
putchar(0xbb);
putchar(0x06);
putchar(0x00);
for(i=0;i<4;i++)
```

```
{
putchar(readmifare_table[i]);
xxor ^= readmifare_table[i];
}
putchar(_block);
xxor ^= _block;
putchar(xxor);
while (!g_bReceCommandOk && !g_bTimeOut);
if (!g_bTimeOut)
{
for(i=0;i<16;i++)
*(ptr+i)=rx_buffer[i+9];
return true;
}
else
return false;

}
//****************
char cmd_writemifare(unsigned char _block, unsigned char *ptr)
{
char i;
unsigned char xxor=0;
```

```
cmd_number=8;rx_index=0;g_bReceCommandOk=0;
g_bTimeOut=0;cont_tim0=0;c_tim0=100;TCCR0=0x03;//start timer0
putchar(0xaa);
putchar(0xbb);
putchar(0x16);
putchar(0x00);
for(i=0;i<4;i++)
{
putchar(writemifare_table[i]);
xxor ^= writemifare_table[i];
}
putchar(_block);xxor ^= _block;
for (i=0;i<16;i++)
{
putchar(*(ptr+i));
xxor ^= *(ptr+i);
};
putchar(xxor);
while (!g_bReceCommandOk && !g_bTimeOut);
if (!g_bTimeOut)
{
return true;
}
```

```
else
return false;
}
//****************
char cmd_halt(void)
{
char i;
unsigned char xxor=0;
cmd_number=9;rx_index=0;g_bReceCommandOk=0;
g_bTimeOut=0;cont_tim0=0;c_tim0=100;TCCR0=0x03;//start timer0
putchar(0xaa);
putchar(0xbb);
putchar(0x05);
putchar(0x00);
for(i=0;i<4;i++)
{
putchar(halt_table[i]);
xxor ^= halt_table[i];
}
putchar(xxor);
while (!g_bReceCommandOk && !g_bTimeOut);
if (!g_bTimeOut)
{
```

```
return true;
}
else
return false;
}
//-----------------------------------------------------------------------------
char connect(char _block)
{
unsigned int tag_type;
unsigned char serial[4];

//Request
if(cmd_Request(0x52, &tag_type))
{
if( cmd_Request(0x52, &tag_type) &&
cmd_anticollision(serial)    &&
cmd_select(serial[0],serial[1],serial[2],serial[3])          &&
cmd_Authentication(0x60,_block,0xff,0xff,0xff,0xff,0xff,0xff) )
{
return true;
};
};
return false;
```

```
};
//-----------------------------------------------------------------------------
```

در فایل Mifare.h اطلاعات خاصی وجود ندارد به همین دلیل از گذاشتن آن در اینجا خودداری کردم .

# فصل پنجم

# خلاصه پروژه و نتایج بدست آمده

تکنولوژی RFID یک تکنولوژی کلیدی برای رسیدن به هدف شبکه اشیا می باشد . در این حال که سیستم های RFID به طور فراگیری گسترش می یابند ، زندگی روزمره ما به این تکنولوژی وابسته می شود . پس کارایی و قابلیت اطمینان بالا مورد نیاز است . در نتیجه مشکل تداخل تگ ها توجه محققان را همچنان جذب می کند . تا به امروز بیشتر شرکت ها برای حل این مشکل از روش های مبتنی بر TDMA استفاده کرده اند . اما با توجه به طبیعت تکنولوژی TDMA ، فقط یک شماره سریال تگ در یک شکاف زمانی قابل بازیابی است . پس برای بهبود سریع کارایی سیستم ، مانند دیگر سیستم های مخابراتی مدرن ، استفاده همزمان از دیگر روش ها مورد نیاز است . تا به امروز بیشتر مقالات بر روی یک روش خاص متمرکز شده اند . برای بهینه سازی بازدهی سیستم ، تکنولوژی های پیشرفته تری مانند تکنولوژی بهینه سازی cross-layer مورد نیاز است . یک سیستم RFID غیر فعال ، برعکس دیگر سیستم های مخابراتی ، یک ساختار نامتوازن بین تگ و ریدر دارد ، جایی که یک تگ غیر فعال محدودیت های توان و قیمت دارد و بنابراین

فقط توانایی انجام الگوریتم های با پیچیدگی کم را دارد . این نتایج بدست آمده سیستم های RFID غیر فعال در طراحی یک طرح ضد تداخل ، یک مشکل منحصر به فرد و چالش برانگیز می باشد .

متاسفانه اطلاعات درباره سیستم های RFID ضدتداخل که به صورت مدارات عملی باشد به دلیل تازگی و به روز بودن مطلب در انحصار چند شرکت خاص بوده و آنها نیز فقط به صورت محصول کامل و پکیج شده آن را در اختیار دیگران قرار می دهند .

سرانجام برای انجام پروژه ای مانند این رو به محصولات مایفر که برد کمی دارند آوردم و یک نمونه از کاربرد های آن را نشان دادم .

**فهرست منابع**


- InTech The approaches in solving passive rfid tag collision problems Hsin-Chin Liu Taiwan 2010
- Tag Anti-collision Algorithms in RFID Systems MAJID ALOTAIBI, ADAM POSTULA, and MARIUS PORTMANN
- www.aftab.ir
- International Standard