802.3ca style conventions

Glen Kramer, Broadcom
Duane Remein, Huawei
Marek Hajduczenia, Charter
Mark Laubach, Broadcom

Motivation and P802.3ca specifics

- P802.3ca draft makes extensive use of
  - Messages (MPCPDUs and CCPDUs) that define multiple fields
  - State diagrams that define variables, constants, and functions
  - Programming code
- When an above element is referenced in text, it is important to differentiate such element from the direct meaning of the word used as its name.
- The IEEE style manual and the 802.3 template do not address the styles for representation of the above elements.
- Lacking the clear, agreed-upon convention, in every draft review cycle, we have style-related comments that propose contradictory changes.

Style for variables (in text)

- **Naming**
  - First letter capitalized. In multi-word names, every word starts with a capital letter.
  - No underscores
  - Examples: *Registered*, *Back2BackEnvelopes*

Font	
Name	Times New Roman
Size	Body text
Style	Italics

- **Exclusions**
  - A variable that starts with a single letter term, can start with the lower case
    - Examples: *wRow*, *rCol*, *xIndex*
  - Variables defined outside of 802.3ca
  - Variable names in the definition headers are not italicized

ClkIn

TYPE: Boolean

The clear on read variable *ClkIn* is set to true on each rising edge of the xMII clock.

Style for constants (in text)

- **Naming**
  - All capital letters. In multi-word names, words are separated using underscores
  - Examples: *SCRAMBLED*, *FEC_PARITY_SIZE*

Font	
Name	Times New Roman
Size	Body text
Style	Italics

- **Exclusions**
  - Constants defined outside of 802.3ca
  - Constant names in the definition headers are not italicized

RATE_ADJ_SIZE

TYPE: integer

The *RATE_ADJ_SIZE* constant represents the number of EQs within the *ADJ_BLOCK_SIZE* block during which the MAC transmission is deferred. The effective MAC rate is equal to <nominal MAC rate> × (1-*RATE_ADJ_SIZE* / *ADJ_BLOCK_SIZE*).

Style for message (PDU) names

- **Naming**
  - All capital letters.
    In multi-word names,
    word are separated
    using underscores
  - Examples: REGISTER_ACK, CC_RESPONSE

Font	
Name	Times New Roman
Size	Body text
Style	None

Style for message field names

- **Naming**
  - First letter capitalized. In multi-word names, every word starts with a capital letter.
  - No underscores
  - Examples: *Timestamp*, *StartTime*, *DiscoveryInfo*
  - Field names can be used directly in the state diagrams, just like variables. They don't need to be redefined in the "Variables" section.

Font	
Name	Times New Roman
Size	Body text
Style	Italics

- **Exclusions**
  - Field names defined outside of 802.3ca
  - Field names in the definition headers are not italicized

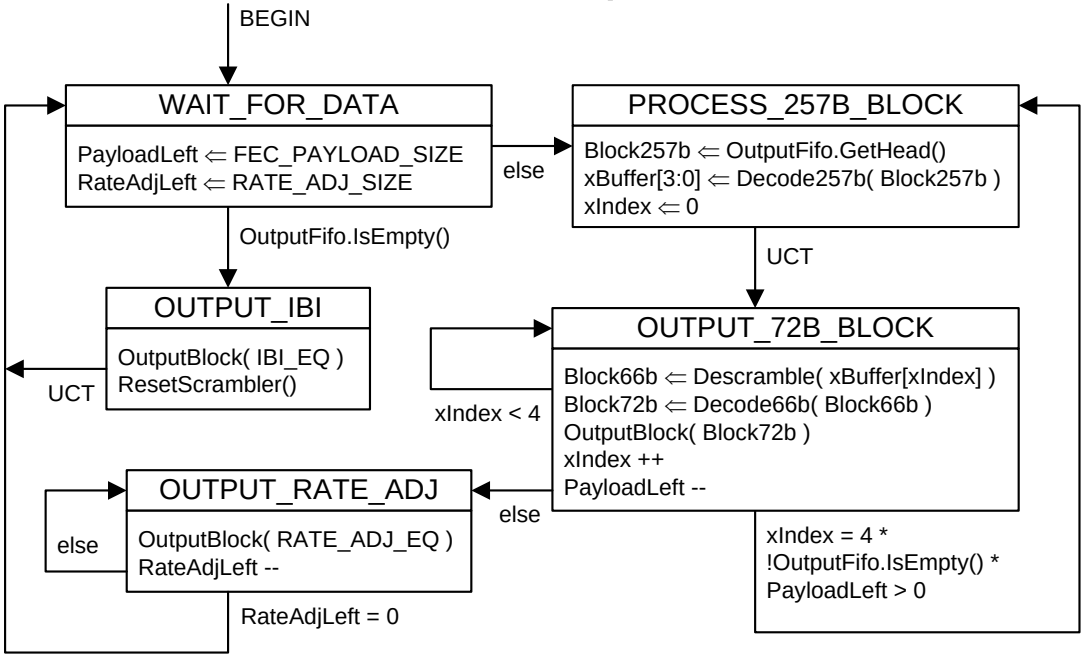
g) DiscoveryInfo. This is a 16-bit flag register. Table 144–7 presents the internal structure of the *DiscoveryInfo* flag field.

Style for state diagrams

- State names are capitalized and joined by underscores. No spaces.
- Transitions into states preferably enter the state header block
- Transitions out of state preferably leave state body block
- Multiple transitions out of several states may be combined into a single transition into a state
- Transitions shall not cross
- Transition length shall be minimized. Long transitions with multiple changes of direction shall be avoided.
- An SD shall have a balanced look and feel, with no parts being too crowded, while other parts remain sparse.

Font	
Name	Arial or Arial monospace
Size	8-10
Style	None

SD Example



Format for Definitions

- Base the format for definitions of constants, variables, and functions on attribute format used in IEEE1904

- **Constant definitions**

CONSTANT_NAME

Type: typename

Description: Description text goes here. Can use multiple lines. Can use multiple lines. Can use multiple lines. Can use multiple lines. Can use multiple lines. Can use multiple lines. Can use multiple lines. Can use multiple lines. Can use multiple lines.

Value: value

- **Variable Definitions**

VariableName

Type: typename

Description: Description text goes here. Can use multiple lines. Can use multiple lines. Can use multiple lines. Can use multiple lines. Can use multiple lines. Can use multiple lines. Can use multiple lines. Can use multiple lines. Can use multiple lines.

- Where appropriate, may use additional categories, such as “Unit” or “Range”

Style for programming code

- C/C++ style pseudocode
- Focus on clarity

Font	
Name	Courier New
Size	Body text
Style	None

Example

EnvContHeader(wCol)

The *EnvContHeader()* function returns a new envelope header with the *EnvType* flag equal to 0, indicating that it is a continuation of the current envelope.

```
EQ EnvContHeader(int col)
{
    EQ hdr;
    hdr<0:7> = 0x80;                //Control bits (1000-0000b)
    hdr<8:15> = 0xFB;               //S-character
    hdr<16> = 0;                    //EnvType identifies ECH
    hdr<18:39> = EnvLeft[col];      //EnvLength
    hdr<40:45> = EnvPam;            //EPAM
    hdr<48:63> = LinkId[col];       //LLID
    hdr<64:71> = CRC8(hdr<8:63>); //Calculate CRC8
    return hdr;
}
```

Emphasis

- It may be necessary or desirable to emphasize various special terms that are not field names, variables, constants, etc.
- Such emphasis is shown by italicizing the word(s) representing the special term.
- Emphasis of special terms should be applied judiciously

16.4 Warnings and cautions

Warnings call attention to the use of materials, processes, methods, procedures, or limits that have to be followed precisely to avoid injury or death.

Cautions call attention to methods and procedures that have to be followed to avoid damage to equipment. A warning is more important
warning shall precede the cau

12.1 Quantity

The word *quantity* has many meanings; in this subclause, the word refers to physical quantities, which are described in units of measure such as length, mass, time, and temperature. A unit is a particular sample of a quantity, chosen so that an appropriate value may be specified. Meter, kilogram, hour, and degree Celsius are some of the units used for the four quantities noted previously.

The value of a quantity is generally expressed as the product of a number and a unit. Quantities and units may be represented in text by letter symbols, and are always so represented in equations. If a number and unit cannot be identified for a quantity, the discussion may concern an amount rather than a quantity, in which case the term *amount* should be used.

Excerpted text is from the IEEE style manual

Motion

- **Accept 802.3ca draft style guidelines as shown in kramer_3ca_4a_0119.pdf, slides 3-10.**

Moved: G. Kramer

Second:

(Procedural, > 50%)

Y: N: A:

Thank You