

Basic General-Purpose Data Structures Used by All Cabinets

Pseudo-code representations of the data structures used by all of the cabinets to process general-purpose data packets are as follows:

```
Enum packet_type ' The type of incoming/outgoing packet
  general_comms = $00; ' $ = hexadecimal
  config_mode_1 = $01;
  config_mode_2 = $02;
  config_request = $03;
  config_response = $04
  special_comms = $05;
End Enum
```

```
Enum unit_type ' The type of cabinet that generated the packet
  control_logic = %0000; ' % = binary
  alu_logic = %0001;
  status_logic = %0010;
  address_logic = %0011;
  rom_unit = %0100;
  ram_unit = %0101;
  input_ports = %0110;
  output_ports = %0111;
  unused_8 = %1000;
  unused_9 = %1001;
  unused_a = %1010;
  unused_b = %1011;
  unused_c = %1100;
  complete_cpu = %1101;
  internet_bridge = %1110;
  pc_bridge = %1111;
End Enum
```

```
Struct info_type ' The type of information being conveyed in the packet
  Bit address; ' Bit 0 of Info Type
  Bit data; ' Bit 1 of Info Type
  Bit status; ' Bit 2 of Info Type
  Bit control; ' Bit 3 of Info Type
  Bit instruction; ' Bit 4 of Info Type
  Bit cpu_signals; ' Bit 5 of Info Type
  Bit spare_6
  Bit spare_7
End Struct
```

```
Struct cpu_signals ' The values on the primary CPU control signals
  Bit iack; ' Bit 0 of CPU Signals
  Bit irq; ' Bit 1 of CPU Signals
  Bit reset; ' Bit 2 of CPU Signals
  Bit busy; ' Bit 3 of CPU Signals
  Bit wr; ' Bit 4 of CPU Signals
  Bit rd; ' Bit 5 of CPU Signals
  Bit spare_6
  Bit spare_7
End Struct
```

```
Struct status ' The values of the status flags
  Bit c; ' Bit 0 = carry flag
  Bit z; ' Bit 1 = zero flag
  Bit n; ' Bit 2 = negative flag
```

```

    Bit o;          ' Bit 3 = overflow flag
    Bit i;          ' Bit 4 = interrupt mask flag
    Bit spare_5
    Bit spare_6
    Bit spare_7
End Struct

Struct control_word ' The various bits in the control word
    5-Bit inst;     ' Bits 0-4 = inst[4:0]           ALU Logic
    Bit clk_alu_acc ' Bit 5 = ACC Clock           ALU Logic
    Bit clk_alu_tmp ' Bit 6 = TMP Clock           ALU Logic
    Bit ena_alu_tbuf ' Bit 7 = Tri-state enable       ALU Logic

    Bit clk_status_i ' Bit 8 = Clock for I bit       Status logic
    Bit clk_status_o ' Bit 9 = Clock for O bit       Status logic
    Bit clk_status_n ' Bit 10 = Clock for N bit      Status logic
    Bit clk_status_z ' Bit 11 = Clock for Z bit      Status logic
    Bit clk_status_c ' Bit 12 = Clock for C bit      Status logic
    Bit set_im       ' Bit 13 = Set interrupt mask   Status Logic
    Bit clr_im       ' Bit 14 = Clear interrupt mask  Status Logic
    Bit sel_status_2_1 ' Bit 15 = Mux control         Status Logic
    Bit ena_status_tbuf ' Bit 16 = Tri-state enable   Status Logic

    Bit clk_addr_pc_ms ' Bit 17 = PC MS clock       Address Logic
    Bit clk_addr_pc_ls ' Bit 18 = PC LS clock       Address Logic
    Bit clk_addr_tpca_ms ' Bit 19 = TMP PCA MS clock   Address Logic
    Bit clk_addr_tpca_ls ' Bit 20 = TMP PCA LS clock   Address Logic
    Bit clk_addr_tpcb_ms ' Bit 21 = TMP PCB MS clock   Address Logic
    Bit clk_addr_tpcb_ls ' Bit 22 = TMP PCB LS clock   Address Logic
    Bit clk_addr_sp_ms ' Bit 23 = SP MS clock       Address Logic
    Bit clk_addr_sp_ls ' Bit 24 = SP LS clock       Address Logic
    Bit clk_addr_iv_ms ' Bit 25 = IV MS clock       Address Logic
    Bit clk_addr_iv_ls ' Bit 26 = IV LS clock       Address Logic
    Bit clk_addr_x_ms ' Bit 27 = X MS clock       Address Logic
    Bit clk_addr_x_ls ' Bit 28 = X LS clock       Address Logic
    Bit clk_addr_latch ' Bit 29 = Address latch clock   Address Logic
    Bit sel_addr_2_1_in ' Bit 30 = Input mux control   Address Logic
    Bit sel_addr_2_1_out ' Bit 31 = Output mux control  Address Logic
    3-Bit sel_addr_5_1[2:0] ' Bits 32-34 = Adder mux control Address Logic
    3-Bit sel_addr_6_1[2:0] ' Bits 35-37 = Address mux control Address Logic
    Bit ena_addr_tbuf_ms ' Bit 38 = Tri-state enable (MS) Address Logic
    Bit ena_addr_tbuf_ls ' Bit 39 = Tri-state enable (LS) Address Logic
End Struct

Struct general_data_packet
    Byte of Struct/Enum packet_type; ' $00 = general data packet
    6-Byte mac_address;              ' Six-byte unique MAC address
    Byte of Struct/Enum unit_type;    ' Type of cabinet
    Byte of Struct info_type;         ' Type of information
    Byte of Struct cpu_signals        ' CPU Signals
    2-Byte address                    ' Address[15:0]
    Byte data                          ' Data[7:0]
    Byte of Struct status              ' Status[4:0]
    5-Byte of Struct control_word     ' The main control word
End Struct

Struct general_sub_packet
    Byte of Struct cpu_signals        ' CPU Signals
    2-Byte address                    ' Address[15:0]

```

```
Byte data           ' Data[7:0]
Byte of Struct status ' Status[4:0]
5-Byte of Struct control_word ' The main control word
End Struct

Enum busy_type
    timed_delay = 0
    busy_signal = 1
End Enum

Enum which_delay
    delay_1 = 0           ' This will typically be the shorter delay
    delay_2 = 1           ' This will typically be the longer delay
End Enum
```