```
MODULE      CPU
TITLE      'Caltech10 CPU'


" Description:  This the Caltech10 CPU interfacing file where we interface
"               between the PAU, DAU, ALU, and CPU. There are also muxes for
"               the input to the ALU PAu and Data Data Bus
"
" Inputs:         ProgramDB[15..0]  - the 16-bit program data bus
"                 Reset             - system reset signal
"                 Clock             - system clock
"
" I/O:            DataDB[7..0]      - the 8-bit data data bus
"
" Outputs:        ProgramAB[12..0]  - the 13-bit program address bus
"                 DataAB[7..0]      - the 8-bit data address bus
"                 RD                - read signal for the data data bus
"                 WR                - write signal for the data data bus
"                 IO                - memory (0) or I/O (1) is being accessed
"
" Status Outputs: Accum[7..0]       - the 8-bit accumulator
"                 Flags[7..0]       - the 8-bit flag register
"                 XReg[7..0]        - the 8-bit X register
"                 SReg[7..0]        - the 8-bit S register
"
" Internal Signals: ALU      - control signal for ALU input
"                   PMU             - control signal for PMU input
"                   PCS             - control signal for input into Data Data bus
"                   AC               - control signal for input into Data Data bus
"                                      PCS     AC      Ouptut
"                                      0       0       [Flags7, 0, Flags5, 0, Flags3..Flags0]
"                                      0       1       Accum
"                                      1       1       Program Counter High 8 bits
"                                      1       0       Program Counter Low 8 bits



" Revision History:
" 02/15/18   Glen George  Initial Revision
" 03/09/18   David Zheng  Interface modules
" 03/10/18   David Zheng  Moved equations for muxes from Control Unit file
" 03/14/18   David Zheng  Edit connections to include muxes & interfacing
" 03/16/18   David Zheng  Edit flag connections for Jmps

" Pin/Signal Declarations

" Inputs

ProgramDB15..ProgramDB0  pin;    "input  16-bit instruction data bus
Reset                    pin;    "input  system reset signal
Clock            pin;    "input  system clock


" I/O

DataDB7..DataDB0 pin;    "I/O  8-bit Data data bus


" Outputs

ProgramAB12..ProgramAB0  pin;    "output  the 13-bit Program address bus
DataAB7..DataAB0 pin;    "output  the 8-bit Data address bus
RD                       pin;    "output  read signal for the Data data bus
WR                       pin;    "output  write signal for the Data data bus
IO                       pin;    "output  accessing memory (0) or I/O (1)


" Status Outputs

Accum7..Accum0           pin;    "the accumulator
Flags7..Flags0           pin;    "the flags
XReg7..XReg0             pin;    "the X register
SReg7..SReg0             pin;    "the S register

" Intermediates
Offset7..Offset0 node ISTYPE 'COM';       "input  into PMU
AccumIn7..AccumIn0       node ISTYPE 'COM';       "input  system reset signal

" Internal signals
ALU                      node ISTYPE 'COM';       "input  control signal for ALU input
PMU                      node ISTYPE 'COM';       "input  control signal for PMU input
PCS                      node ISTYPE 'COM';       "input  control signal for input into Data Data bus
AC                       node ISTYPE 'COM';       "input  control signal for input into Data Data bus
"                                                  PCS     AC      Ouptut
"                                                  0       0       [Flags7, 0, Flags5, 0, Flags3..Flags0]
"                                                  0       1       Accum
"                                                  1       1       Program Counter High 8 bits
"                                                  1       0       Program Counter Low 8 bits

" declare the used modules
ProgramAccess    INTERFACE  (EigBitIn7..EigBitIn0, ThrBitIn12..ThrBitIn0,
                         RTS, Upper, Abs, Direct, Load, Stop, Reset, Clock
                         -> PCIn12..PCIn0);

DataAccess       INTERFACE  (Offset7..Offset0,
                        SelX, SelS, Post, Inc, Zero, Store, Reset, Clock
                        -> DataAddr7..DataAddr0, X7..X0, S7..S0);

ALUAccess        INTERFACE  (InputB7..InputB0, FOpInD, FOpInC, FOpInB, FOpInA,
                        AddF, Subtract, Store, WithCarry, DAccumStore, DFlagStore,
                        RLC, NOP, Reset, Clock -> AccumOut7..AccumOut0, CarryFlag,
                        OverflowFlag, SignFlag, ZeroFlag, Flag7..Flag4);

ControlUnitAccess INTERFACE  (PD15..PD0, FlagReg7..FlagReg4, ZeroF,
                        Sign, Overflow, Carry, Clock, Reset
                        -> IR15..IR0, RTS, Upper, Abs, Direct, Load, FOpInD, FOpInC, FOpInB, FOpInA,
                        AddF, Subtract, Store_ALU, WithCarry, DAccumStore,
                        DFlagStore, RLCcs, Stop_PAU, Stop_IR, NOPcs, SelX, SelS, Post, Inc, Zero, ALU, PMU, PCS,
                        AC, WR, RD, IO, Store_DMU);
```

```
" create instances of the modules
PAUnit    FUNCTIONAL_BLOCK  ProgramAccess;
DAUnit    FUNCTIONAL_BLOCK  DataAccess;
ALUnit    FUNCTIONAL_BLOCK  ALUAccess;
CPUnit    FUNCTIONAL_BLOCK  ControlUnitAccess;


" Busses

ProgramAB  =  [ProgramAB12..ProgramAB0];"Program address bus
ProgramDB  =  [ProgramDB15..ProgramDB0];"Program data bus
DataAB     =  [DataAB7..DataAB0]; "Data address bus
DataDB     =  [DataDB7..DataDB0]; "Data data bus
IRHigh     =  [CPUnit.IR15..CPUnit.IR8];"Program data bus high 8 bits
IRLow      =  [CPUnit.IR7..CPUnit.IR0];   "Program data bus low 8 bits
Accum      =  [Accum7..Accum0];           "Input into Accumulator
Offset     =  [Offset7..Offset0];"Input into the PMU
Flags      =  [Flags7..Flags0];           "Flags for the PCU
AccumIn    =  [AccumIn7..AccumIn0];       "Input into the Accumulator from a mux
PCL        =  [ProgramAB7..ProgramAB0];   " Low 8 bits of the program counter
PCH        =  [0, 0, 0, ProgramAB12..ProgramAB8]; " High 5 bits of the program counter padded with 0s

EQUATIONS

" Contol Signals for muxes inside PCU outside other modules
ALU     =  CPUnit.ALU;
PMU     =  CPUnit.PMU;
PCS     =  CPUnit.PCS;
AC      =  CPUnit.AC;
"input to ALU & Accumulator
"When ALU is on input low 8 bits of IR
"When ALU low input the Data Data Bus
AccumIn  =       (IRLow    &  ALU) #
                 (DataDB   & !ALU);

"output to the data bus
"When PMU is high input the Data Data Bus
"When PMU is low input the low 8 bits of IR
Offset   =       (IRLow    & !PMU) #
                 (DataDB   &  PMU);

"input to PMU
"0      0       [Flags7, 0, Flags5, 0, Flags3..Flags0]
"0      1       Accum
"1      1       Program Counter High 8 bits
"1      0       Program Counter Low 8 bits
DataDB   =       (PCL     &  PCS & !AC) #
                 (PCH    &  PCS &  AC) #
                 ([Flags7, 0, Flags5, 0, Flags3..Flags0]  & !PCS & !AC) #
                 (Accum   & !PCS &  AC);

" output enables

" Output CPU into Control Unit
CPUnit.[PD15..PD0]       =  [ProgramDB15..ProgramDB0];

"Output from Control unit to input of ALU
ALUnit.FOpInA            =  CPUnit.FOpInA;
ALUnit.FOpInB            =  CPUnit.FOpInB;
ALUnit.FOpInC            =  CPUnit.FOpInC;
ALUnit.FOpInD            =  CPUnit.FOpInD;
ALUnit.Subtract          =  CPUnit.Subtract;
ALUnit.WithCarry =  CPUnit.WithCarry;
ALUnit.Store             =  CPUnit.Store_ALU;
ALUnit.AddF              =  CPUnit.AddF;
ALUnit.DAccumStore       =  CPUnit.DAccumStore;
ALUnit.DFlagStore        =  CPUnit.DFlagStore;
ALUnit.RLC               =  CPUnit.RLCcs;
ALUnit.NOP               =  CPUnit.NOPcs;
ALUnit.[InputB7..InputB0] =  [AccumIn7..AccumIn0];

" Output of ALU into CPU
[Flags7..Flags0] =  ALUnit.[Flag7..Flag4, CarryFlag, OverflowFlag, SignFlag, ZeroFlag];
[Accum7..Accum0] =  ALUnit.[AccumOut7..AccumOut0];

" Output CPU into Control unit
CPUnit.Carry             =  ALUnit.CarryFlag;
CPUnit.Overflow          =  ALUnit.OverflowFlag;
CPUnit.ZeroF             =  ALUnit.ZeroFlag;
CPUnit.Sign              =  ALUnit.SignFlag;
CPUnit.[FlagReg7..FlagReg4]       =  [Flags7..Flags4];

"Output CPU into PAU
PAUnit.RTS               =  CPUnit.RTS;
PAUnit.Upper             =  CPUnit.Upper;
PAUnit.Abs               =  CPUnit.Abs;
PAUnit.Direct            =  CPUnit.Direct;
PAUnit.Load              =  CPUnit.Load;
PAUnit.Stop              =  CPUnit.Stop_PAU;
PAUnit.[EigBitIn7..EigBitIn0]     = Offset;
PAUnit.[ThrBitIn12..ThrBitIn0]    = CPUnit.[IR12..IR0];

"Output PAU into CPU
[ProgramAB12..ProgramAB0] =  PAUnit.[PCIn12..PCIn0];

" Output of CPU into DAU
DAUnit.SelX              =  CPUnit.SelX;
DAUnit.SelS              =  CPUnit.SelS;
DAUnit.Post              =  CPUnit.Post;
DAUnit.Inc               =  CPUnit.Inc;
DAUnit.Zero              =  CPUnit.Zero;
DAUnit.Store             =  CPUnit.Store_DMU;
DAUnit.[Offset7..Offset0] =  CPUnit.[IR7..IR0];

" Output of DAU into CPU
[DataAB7..DataAB0]       =  DAUnit.[DataAddr7..DataAddr0];

"Output of Control Unit into CPU
RD       =  CPUnit.RD;
WR       =  CPUnit.WR;
```

```
IO      =  CPUnit.IO;

" Setting Reset and Clock
PAUnit.Reset     =  Reset;
ALUnit.Reset     =  Reset;
DAUnit.Reset     =  Reset;
CPUnit.Reset     =  Reset;
PAUnit.Clock     =  Clock;
ALUnit.Clock     =  Clock;
DAUnit.Clock     =  Clock;
CPUnit.Clock     =  Clock;

" X and S Reg
[XReg7..XReg0] = DAUnit.[X7..X0];
[SReg7..SReg0] = DAUnit.[S7..S0];

" Data data bus is enabled when writing
DataDB.OE  =  WR;

" most signals are always enabled
ProgramAB.OE         =  ^h1FFF;
DataAB.OE            =  ^hFF;
RD.OE                =  1;
WR.OE                =  1;
IO.OE                =  1;
[Accum7..Accum0].OE  =  ^hFF;
[Flags7..Flags0].OE  =  ^hFF;
[XReg7..XReg0].OE    =  ^hFF;
[SReg7..SReg0].OE    =  ^hFF;


END  CPU
```