```
-- ControlDefs.Mesa  Edited by Sandman on April 10, 1978  10:10 AM

DIRECTORY
  AltoDefs: FROM "altodefs",
  Mopcodes: FROM "mopcodes",
  SegmentDefs: FROM "segmentdefs";

ControlDefs: DEFINITIONS =
  BEGIN

  BYTE: TYPE = AltoDefs.BYTE;

  -- control link definitions

  ControlLinkTag: TYPE = {frame, procedure, indirect, unbound};

  ControlLink: TYPE = MACHINE DEPENDENT RECORD [
    SELECT OVERLAID ControlLinkTag FROM
      frame => [
        frame: FrameHandle],
      procedure => [
        gfi: GFTIndex,
        ep: EPIndex,
        tag: ControlLinkTag],
      indirect => [
        SELECT OVERLAID * FROM
          port => [
            port: PortHandle],
          link => [
            link: POINTER TO ControlLink],
          ENDCASE],
      ENDCASE];

  ProcDesc, SignalDesc: TYPE = procedure ControlLink;

  TrapLink: ControlLink = ControlLink[frame[NullFrame]];

  PortHandle: TYPE = POINTER TO Port;

  Port: TYPE = MACHINE DEPENDENT RECORD [
    SELECT OVERLAID * FROM
      representation => [
        in, out: UNSPECIFIED],
      links => [
        frame: FrameHandle,
        dest: ControlLink],
      ENDCASE];


  -- frame definitions

  MaxFrameSize: CARDINAL = 4096-4;
  MaxSmallFrameSize: CARDINAL = 231;

  FrameVec: ARRAY [0..MaxAllocSlot) OF [0..MaxFrameSize] =
    [7,11,15,19,23,27,31,39,47,55,67,79,95,111,127,147,171,199,231];

  FrameClass: TYPE = {global, local, signal, catch, dying};

  Frame: TYPE = MACHINE DEPENDENT RECORD [
    accesslink: GlobalFrameHandle,
    pc: WordPC,
    returnlink: ControlLink,
    extensions: SELECT OVERLAID FrameClass FROM
      local => [
        unused: UNSPECIFIED,
        local: ARRAY [0..0) OF UNSPECIFIED],
      signal => [
        mark: BOOLEAN,
        unused: [0..77777B]],
      catch => [
        unused: UNSPECIFIED,
        staticlink: FrameHandle,
        messageval: UNSPECIFIED],
      dying => [
        state: {dead, alive},
```

```
      unused: [0..77777B]],
    ENDCASE];

FrameHandle: TYPE = POINTER TO Frame;
NullFrame: FrameHandle = LOOPHOLE[0];

GetReturnLink: PROCEDURE RETURNS [ControlLink] =
  MACHINE CODE BEGIN Mopcodes.zLLB, returnOffset END;
GetReturnFrame: PROCEDURE RETURNS [FrameHandle] = LOOPHOLE [GetReturnLink];
SetReturnLink: PROCEDURE [ControlLink] =
  MACHINE CODE BEGIN Mopcodes.zSLB, returnOffset END;
SetReturnFrame: PROCEDURE [FrameHandle] = LOOPHOLE [SetReturnLink];

GlobalFrame: TYPE = MACHINE DEPENDENT RECORD [
  gfi: GFTIndex,
  unused: [0..1], -- reserved for future gfi expansion
  copied, alloced, shared, started: BOOLEAN,
  trapxfers, codelinks: BOOLEAN,
  code: FrameCodeBase,
  codesegment: SegmentDefs.FileSegmentHandle,
  global: ARRAY [0..0) OF UNSPECIFIED];

FrameCodeBase: TYPE = MACHINE DEPENDENT RECORD [
  SELECT OVERLAID * FROM
    in => [
      SELECT OVERLAID * FROM
        codebase => [
          codebase: POINTER],
        prefix => [
          prefix: POINTER TO CSegPrefix],
        ENDCASE],
    out => [
      offset: CARDINAL],
    either => [
      fill: [0..77777B],
      swappedout: BOOLEAN],
    ENDCASE];

GlobalFrameHandle: TYPE = POINTER TO GlobalFrame;
NullGlobalFrame: GlobalFrameHandle = LOOPHOLE[0];

Alloc: PROCEDURE [CARDINAL] RETURNS [POINTER] =
  MACHINE CODE BEGIN Mopcodes.zALLOC END;
Free: PROCEDURE [POINTER] =
  MACHINE CODE BEGIN Mopcodes.zFREE END;


-- The following offsets are used by the compiler and MUST
-- reflect the field offsets in the definitions of Frame

-- local frames
accessOffset: CARDINAL = 0;
pcOffset: CARDINAL = 1;
returnOffset: CARDINAL = 2;
-- global frames
gfiOffset: CARDINAL = 0;
codebaseOffset: CARDINAL = 1;

-- efficiently addressable portion of frames

globalbase: CARDINAL = SIZE[GlobalFrame];
globalslots: CARDINAL = 8;
localbase: CARDINAL = SIZE[local Frame];
localslots: CARDINAL = 8;
framelink: CARDINAL = localbase;
lprocslots, procslots: CARDINAL = 16;

-- code segments

WordPC: TYPE = RECORD [INTEGER];
BytePC: TYPE = RECORD [CARDINAL];

InstWord: TYPE = MACHINE DEPENDENT RECORD [
  oddbyte, evenbyte: BYTE];

DInstWord: TYPE = MACHINE DEPENDENT RECORD [
```

```
    evenbyte, oddbyte: BYTE];

  FieldDescriptor: TYPE = MACHINE DEPENDENT RECORD [
    offset: BYTE, posn: [0..16), size: [1..16]];

  EPRange: CARDINAL = 32;
  EPIndex: TYPE = [0..EPRange);

  -- Global frame size preceeds body zero

  CSegPrefix: TYPE = MACHINE DEPENDENT RECORD [
    swapinfo: WORD,
    stops: BOOLEAN,
    fill: [0..37B],
    ngfi: [1..MaxNGfi],
    nlinks: [0..MaxNLinks],
    entry: ARRAY [0..0) OF EntryVectorItem];

  MaxNLinks: CARDINAL = 255;

  -- if framesize is MaxAllocSlot, actual framesize is at offset initialpc-1.

  EntryVectorItem: TYPE = MACHINE DEPENDENT RECORD [
    initialpc: WordPC,
    defaults: BOOLEAN,
    nparams: [0..177B],
    framesize: [0..377B]];

  MainBodyIndex: CARDINAL = 0;


  -- Global Frame Table definitions

  MaxNGfi: CARDINAL = 4;

  GFT: POINTER TO ARRAY [0..0) OF GFTItem = LOOPHOLE[1400B];

  GFTItem: TYPE = MACHINE DEPENDENT RECORD [
    frame: GlobalFrameHandle,
    epbase: CARDINAL];

  GFTIndex: TYPE = [0..777B];
  GFTNull: GFTIndex = LAST[GFTIndex];
  NullEpBase: CARDINAL = LAST[CARDINAL];

  MaxGFTSize: CARDINAL = (LAST[GFTIndex]+1)*SIZE[GFTItem];


  -- system frame allocation vector

  AllocTag: TYPE = {frame, empty, indirect, enable};

  AVItem: TYPE = MACHINE DEPENDENT RECORD [
    SELECT OVERLAID * FROM
      data => [
        fsi: [0..37777B],
        tag: AllocTag],
      link => [
        link: POINTER TO AVItem],
      frame => [
        frame: FrameHandle],
      ENDCASE];

  MaxAllocSlot: CARDINAL = 19;
  LargeReturnSlot: CARDINAL = MaxAllocSlot+1;
  SpecialReturnSlot: CARDINAL = MaxAllocSlot+2;
  AllocationVectorSize: CARDINAL = 22;
  AllocationVector: TYPE = ARRAY [0..AllocationVectorSize) OF AVItem;
  AV: POINTER TO AllocationVector = LOOPHOLE[1000B];


  -- control registers

  -- known by the microcode
  WDCreg: CARDINAL = 1;          -- wakeup disable counter
  XTSreg: CARDINAL = 2;          -- xfer trap status
```

```
  XTPreg: CARDINAL = 3;          -- xfer trap parameter
  ATPreg: CARDINAL = 4;          -- alloc trap parameter
  OTPreg: CARDINAL = 5;          -- other trap parameter
  CLKreg: CARDINAL = 6;          -- real time clock low bits

  -- known by compiler only
  Lreg: CARDINAL = 375B;         -- local frame
  Greg: CARDINAL = 376B;         -- global frame
  Creg: CARDINAL = 377B;         -- code base

  MaxParmsInStack: CARDINAL = 5;

  SVPointer: TYPE = POINTER TO StateVector;

  StateVector: TYPE = MACHINE DEPENDENT RECORD [
    stk: ARRAY[0..7] OF UNSPECIFIED,
    instbyte: BYTE,
    fill: [0..17B],
    stkptr: [0..17B],  -- 0 => empty stack
    dest, source: UNSPECIFIED];


  -- system transfer vector

  SD: POINTER TO ARRAY [0..0) OF UNSPECIFIED = LOOPHOLE[1060B];

  END.
```