Appendix 4: The Hack Chip Set

The chips are sorted alphabetically, by name. In the on-line version of this document, this API format comes handy: To use a chip-part, copy-paste the chip signature into your HDL program, then fill-in the missing bindings (also called "connections").

```
Add16(a= ,b= ,out= )  /* Adds up two 16-bit two's complement values */
ALU(x= ,y= ,zx= ,nx= ,zy= ,ny= ,f= ,no= ,out= ,zr= ,ng= )  /* Hack ALU */
And(a= ,b= ,out= )  /* And gate */
And16(a= ,b= ,out= )  /* 16-bit And */
ARegister(in= ,load= ,out= )  /* Address register (built-in) */
Bit(in= ,load= ,out= )  /* 1-bit register */
CPU(inM= ,instruction= ,reset= ,outM= ,writeM= ,addressM= ,pc= )  /* Hack CPU */
DFF(in= ,out= )  /* Data flip-flop gate (built-in) */
DMux(in= ,sel= ,a= ,b= )  /* Channels the input to one out of two outputs */
DMux4Way(in= ,sel= ,a= ,b= ,c= ,d= )  /* Channels the input to one out of four outputs */
DMux8Way(in= ,sel= ,a= ,b= ,c= ,d= ,e= ,f= ,g= ,h= )  /* Channels the input to one out of eight outputs */
DRegister(in= ,load= ,out= )  /* Data register (built-in) */
HalfAdder(a= ,b= ,sum= , carry= )  /* Adds up 2 bits */
FullAdder(a= ,b= ,c= ,sum= ,carry= )  /* Adds up 3 bits */
Inc16(in= ,out= )  /* Sets out to in + 1 */
Keyboard(out= )  /* Keyboard memory map (built-in) */
Memory(in= ,load= ,address= ,out= )  /* Data memory of the Hack platform (RAM) */
Mux(a= ,b= ,sel= ,out= )  /* Selects between two inputs */
Mux16(a= ,b= ,sel= ,out= )  /* Selects between two 16-bit inputs */
Mux4Way16(a= ,b= ,c= ,d= ,sel= ,out= )  /* Selects between four 16-bit inputs */
Mux8Way16(a= ,b= ,c= ,d= ,e= ,f= ,g= ,h= ,sel= ,out= )  /* Selects between eight 16-bit inputs */
Nand(a= ,b= ,out= )  /* Nand gate (built-in) */
Not16(in= ,out= )  /* 16-bit Not */
Not(in= ,out= )  /* Not gate */
Or(a= ,b= ,out= )  /* Or gate */
Or8Way(in= ,out= )  /* 8-way Or */
Or16(a= ,b= ,out= )  /* 16-bit Or */
PC(in= ,load= ,inc= ,reset= ,out= )  /* Program Counter */
RAM8(in= ,load= ,address= ,out= )  /* 8-word RAM */
RAM64(in= ,load= ,address= ,out= )  /* 64-word RAM */
RAM512(in= ,load= ,address= ,out= )  /* 512-word RAM */
RAM4K(in= ,load= ,address= ,out= )  /* 4K RAM */
RAM16K(in= ,load= ,address= ,out= )  /* 16K RAM */
Register(in= ,load= ,out= )  /* 16-bit register */
ROM32K(address= ,out= )  /* Instruction memory of the Hack platform (ROM, built-in) */
Screen(in= ,load= ,address= ,out= )  /* Screen memory map (built-in) */
Xor(a= ,b= ,out= )  /* Xor gate */
```