Table 1: The Anitra *CPU* Hardware Registers and Signals List

Symbol	Full Name	Description
Internal Registers		
A	Address register	Fetches *near addresses* from data bus
S	Segment register	Fetches or clears current *segment number*
PC	Program counter	Counts 8 most significant bits of the *instruction pointer*
R	Accumulator result	Stores a value for arithmetic addition
LP	End of loop detector	Used to detect whether current instruction is last in *loop*
BR	Branch if carry detector	Used to fetch any carry signal during accumulator addition
ES	Execution stepper	Determines and holds the current *execution step*
External Input Signals		
CK	System clock	Makes Anitra proceed to next execution step on positive edge
_R	System reset	Makes Anitra reset all of its internal registers when low
_DAD	Debug fetch A	Clocks A on negative edge during *debug* , should otherwise be high
DSG	Debug fetch S	Clocks S on positive edge during *debug* , should otherwise be high
External Output Signals		
A[0..14]	Memory address bus	Holds current address to be selected by memory
WRM	Write enable memory	Requests memory write when high
DOK	Debug possibility notification	*Debug* possible when high
RET	Loop return notification	Shutdown possible when high
IOX	IO exchange request	Requests *IO exchange* on positive edge/when high[†]
External Bidirectional Signals		
D[0..7]	Data bus	Conveys data between *CPU* , Mem and *external interface* (I/O)
External Other Signals		
VCC	Power	Power (+5V)
GND	Ground	Ground
Internal Signals to Registers		
_CLS	Clear S	Clears S when low
_CLR	Clear R	Clears R when low
CPA	Clear lower PC	Clears PC's 4 least significant bits when high
CPB	Clear upper PC	Clears PC's 4 most significant bits when high
_IPA	Increase lower PC	Increases PC's lower 4 bit counter on negative edge
_IPB	Increase upper PC	Increases PC's upper 4 bit counter on negative edge
_OEA	Output enable A	Tri-state activates A when low
_OEP	Output enable PC	Tri-state activates PC when low
_OER	Output enable R	Tri-state activates R when low
FEA	Fetch A	Clocks A on positive edge
FES	Fetch S	Clocks S on positive edge
FER	Fetch R	Clocks R on positive edge
PCL	Instruction pointer lower bit	Defines *instruction pointer* 's least significant bit
SLO	Segment lower bit out	Overrides current *segment number* 's least significant bit
_ESR	Reset ES	Resets ES to first *execution step* when low
_LPR	Reset LP	Resets LP when low
_BRR	Reset BR	Resets BR when low
LPK	Clock LP	Clocks LP on positive edge
BRK	Clock BR	Clocks BR on positive edge
Internal Signals from Registers		
CAR	Accumulator carry	Holds *accumulator* 's carry signal
ARQ	PC argument Q denoting bit	High when current *argument* is the last in instruction
BLK	PC block change denoting bit	Negative edge means next instruction is first in block
LOP	PC loop start denoting bit	Negative edge means next instruction is first in loop
INQ	Instruction qualifier bit	May hold current instruction's *instruction qualifier bit*
SLI	Segment lower bit in	Holds least significant bit fetched in S
E[0..3]	Execution step	Set high in turn according to the current *execution step*
_E[0..3]	Execution step complement	Set low in turn according to the current *execution step*
LPQ	LP status	May be high when current instruction is last in *loop*
_LPQ	LP status complement	May be low when current instruction is last in *loop*
BRQ	BR status	May hold carry of last *accumulator* operation
Internal Bus		
R[0..7]	Accumulator result bus	Holds current *accumulator* value

[†] The data bus will be in the input state when IOX is high. For *IO exchange* , an output value should be fetched from the data bus on the positive edge of IOX, and an input value should be held on the data bus throughout the high time of IOX.

Note: An underscore (_) in a signal name denotes an active-low or negative edge-triggered signal. B[0..n] denotes a bus B of width *n* bits.