Gem

Generated by Doxygen 1.8.1.2

Wed Mar 4 2015 22:55:33

Contents

Chapter 1

Class Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

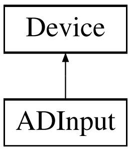
Chapter 4

Class Documentation

4.1 ADInput Class Reference

```
#include <ADInput.h>
```

Inheritance diagram for ADInput:



Public Member Functions

- ADInput (const char ∗name)
- ∼ADIxInput ()
- byte peek (int addr)
- void poke (int addr, byte b)
- int getSize ()
- bool getAddressName (char ∗str, int len, int addr)

Additional Inherited Members

4.1.1 Constructor & Destructor Documentation

4.1.1.1 ADInput::ADInput (const char ∗ *name*)

4.1.1.2 ADInput::∼ADIxInput ( )

4.1.2 Member Function Documentation

4.1.2.1 bool ADInput::getAddressName (char ∗ *str,* int *len,* int *addr*) `[virtual]`

Reimplemented from Device.

4.1.2.2 int ADInput::getSize ( )

Reimplemented from Device.

4.1.2.3 byte ADInput::peek (int *addr*)

Reimplemented from Device.

4.1.2.4 void ADInput::poke (int *addr,* byte *b*)

Reimplemented from Device.

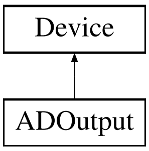
The documentation for this class was generated from the following files:

- ADInput.h
- ADInput.cpp

4.2 ADOutput Class Reference

```
#include <ADOutput.h>
```

Inheritance diagram for ADOutput:



Public Member Functions

- ADOutput ()
- ∼ADIxOutput ()
- byte peek (int addr)
- void poke (int addr, byte b)
- int getSize ()

Additional Inherited Members

4.2.1 Constructor & Destructor Documentation

4.2.1.1 ADOutput::ADOutput ( )

4.2.1.2 ADOutput::∼ADIxOutput ( )

4.2.2 Member Function Documentation

4.2.2.1 int ADOutput::getSize ( )

Reimplemented from Device.

4.2.2.2 byte ADOutput::peek (int *addr*)

Reimplemented from Device.

4.2.2.3 void ADOutput::poke (int *addr,* byte *b*)

Reimplemented from Device.

The documentation for this class was generated from the following files:

- ADOutput.h
- ADOutput.cpp

4.3 CodeWindow Class Reference

```
#include <CodeWindow.h>
```

Public Member Functions

- CodeWindow (int rows, int cols, int atRow, int atCol, R6502 ∗cpu)
- ∼CodeWindow ()
- void display (int addr)

4.3.1 Constructor & Destructor Documentation

4.3.1.1 CodeWindow::CodeWindow (int *rows,* int *cols,* int *atRow,* int *atCol,* R6502 ∗ *cpu*)

4.3.1.2 CodeWindow::∼CodeWindow ( )

4.3.2 Member Function Documentation

4.3.2.1 void CodeWindow::display (int *addr*)

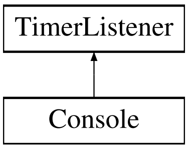
The documentation for this class was generated from the following files:

- CodeWindow.h
- CodeWindow.cpp

4.4 Console Class Reference

```
#include <Console.h>
```

Inheritance diagram for Console:



Public Member Functions

- Console (R6502 ∗proc)
- ∼Console ()
- void initScreen ()

- void updateScreen ()
- void updateHex ()
- void updateStatus ()
- void updateCode ()
- void commandLoop ()
- void timerFired ()

4.4.1 Constructor & Destructor Documentation

4.4.1.1 Console::Console (R6502 ∗ *proc*)

4.4.1.2 Console::∼Console ( )

4.4.2 Member Function Documentation

4.4.2.1 void Console::commandLoop ( )

4.4.2.2 void Console::initScreen ( )

4.4.2.3 void Console::timerFired ( ) [virtual]

Reimplemented from TimerListener.

4.4.2.4 void Console::updateCode ( )

4.4.2.5 void Console::updateHex ( )

4.4.2.6 void Console::updateScreen ( )

4.4.2.7 void Console::updateStatus ( )

The documentation for this class was generated from the following files:

- Console.h
- Console.cpp

4.5 CPUState Class Reference

```
#include <CPUState.h>
```

Public Member Functions

- CPUState (MemoryMap ∗mem)
- ∼CPUState ()
- void reset ()
- int load (int addr)
- byte getStatusFlag ()
- void getStatusFlagAsString (char ∗str, int len)
- void disassembleOp (char ∗str, int len)

Public Attributes

- byte a
- byte x
- byte y
- byte sp
- word pc
- byte ir
- byte args [2]
- word lastPc
- int instSize
- bool opTrap
- bool irqAsserted
- bool nmiAsserted
- bool carryFlag
- bool negativeFlag
- bool zeroFlag
- bool irqDisableFlag
- bool decimalModeFlag
- bool breakFlag
- bool overflowFlag
- long stepCounter
- bool running

4.5.1 Constructor & Destructor Documentation

4.5.1.1 CPUState::CPUState (MemoryMap ∗ *mem*)

4.5.1.2 CPUState::∼CPUState ( )

4.5.2 Member Function Documentation

4.5.2.1 void CPUState::disassembleOp (char ∗ *str,* int *len*)

4.5.2.2 byte CPUState::getStatusFlag ( )

4.5.2.3 void CPUState::getStatusFlagAsString (char ∗ *str,* int *len*)

4.5.2.4 int CPUState::load (int *addr*)

4.5.2.5 void CPUState::reset ( )

4.5.3 Member Data Documentation

4.5.3.1 byte CPUState::a

4.5.3.2 byte CPUState::args[2]

4.5.3.3 bool CPUState::breakFlag

4.5.3.4 bool CPUState::carryFlag

4.5.3.5 bool CPUState::decimalModeFlag

4.5.3.6 int CPUState::instSize

4.5.3.7 byte CPUState::ir

4.5.3.8 bool CPUState::irqAsserted

4.5.3.9 bool CPUState::irqDisableFlag

4.5.3.10 word CPUState::lastPc

4.5.3.11 bool CPUState::negativeFlag

4.5.3.12 bool CPUState::nmiAsserted

4.5.3.13 bool CPUState::opTrap

4.5.3.14 bool CPUState::overflowFlag

4.5.3.15 word CPUState::pc

4.5.3.16 bool CPUState::running

4.5.3.17 byte CPUState::sp

4.5.3.18 long CPUState::stepCounter

4.5.3.19 byte CPUState::x

4.5.3.20 byte CPUState::y

4.5.3.21 bool CPUState::zeroFlag

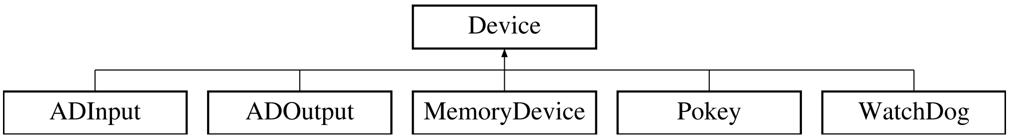
The documentation for this class was generated from the following files:

- CPUState.h
- CPUState.cpp

4.6 Device Class Reference

```
#include <Device.h>
```

Inheritance diagram for Device:



Public Member Functions

- Device ()
- Device (const char ∗newName)
- const char ∗ getName ()

- void setName (const char ∗newName)
- virtual bool getAddressName (char ∗str, int len, int addr)
- int getSize ()
- void setSize (int newSize)
- byte peek (int addr)
- void poke (int addr, byte b)
- virtual byte readByte (int addr)
- virtual void writeByte (int addr, byte b)
- bool save (const char ∗loc, bool overwrite)
- int load (const char ∗loc)
- void setListener (DeviceListener ∗lstnr)
- void fireReadListener (int addr)
- void fireWriteListener (int addr, byte val)

Protected Attributes

- char name [MAXDEVNAME]
- DeviceListener ∗ listener
- int size

4.6.1 Constructor & Destructor Documentation

4.6.1.1 Device::Device ( )

4.6.1.2 Device::Device (const char ∗ *newName*)

4.6.2 Member Function Documentation

4.6.2.1 void Device::fireReadListener (int *addr*)

4.6.2.2 void Device::fireWriteListener (int *addr,* byte *val*)

4.6.2.3 bool Device::getAddressName (char ∗ *str,* int *len,* int *addr*) [virtual]

Reimplemented in WatchDog, Pokey, and ADInput.

4.6.2.4 const char∗ Device::getName ( ) [inline]

Reimplemented in MemoryDevice, WatchDog, and Pokey.

4.6.2.5 int Device::getSize ( ) [inline]

Reimplemented in MemoryDevice, WatchDog, Pokey, ADInput, and ADOutput.

4.6.2.6 int Device::load (const char ∗ *loc*)

4.6.2.7 byte Device::peek (int *addr*)

Reimplemented in WatchDog, Pokey, MemoryDevice, ADInput, and ADOutput.

4.6.2.8 void Device::poke (int *addr,* byte *b*)

Reimplemented in WatchDog, Pokey, MemoryDevice, ADInput, and ADOutput.

4.6.2.9 virtual byte Device::readByte (int *addr*) [virtual]

4.6.2.10 bool Device::save (const char ∗ *loc,* bool *overwrite*)

4.6.2.11 void Device::setListener (DeviceListener ∗ *lstnr*)

4.6.2.12 void Device::setName (const char ∗ *newName*)

4.6.2.13 void Device::setSize (int *newSize*) [inline]

4.6.2.14 virtual void Device::writeByte (int *addr,* byte *b*) [virtual]

4.6.3 Member Data Documentation

4.6.3.1 DeviceListener∗ Device::listener [protected]

4.6.3.2 char Device::name[MAXDEVNAME] [protected]

4.6.3.3 int Device::size [protected]

The documentation for this class was generated from the following files:

- Device.h
- Device.cpp

4.7 DeviceListener Class Reference

`#include <DeviceListener.h>`

Public Member Functions

- virtual void readListener (Device ∗dev, int addr)
- virtual void writeListener (Device ∗dev, int addr, byte val)

4.7.1 Member Function Documentation

4.7.1.1 void DeviceListener::readListener (Device ∗ *dev,* int *addr*) [virtual]

4.7.1.2 void DeviceListener::writeListener (Device ∗ *dev,* int *addr,* byte *val*) [virtual]

The documentation for this class was generated from the following files:

- DeviceListener.h
- DeviceListener.cpp

4.8 Machine Class Reference

`#include <Machine.h>`

Public Member Functions

- Machine ()
- virtual void configureDevides ()
- virtual void configureProcessor ()

Protected Attributes

- Processor ∗ proc
- MemoryMap ∗ memory

4.8.1 Constructor & Destructor Documentation

4.8.1.1 Machine::Machine ( )

4.8.2 Member Function Documentation

4.8.2.1 virtual void Machine::configureDevides ( ) [virtual]

4.8.2.2 void Machine::configureProcessor ( ) [virtual]

4.8.3 Member Data Documentation

4.8.3.1 MemoryMap∗ Machine::memory [protected]

4.8.3.2 Processor∗ Machine::proc [protected]

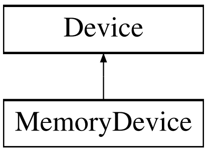
The documentation for this class was generated from the following files:

- Machine.h
- Machine.cpp

4.9 MemoryDevice Class Reference

```
#include <MemoryDevice.h>
```

Inheritance diagram for MemoryDevice:



Public Member Functions

- MemoryDevice (int size_in_bytes, short int ro)
- ∼MemoryDevice ()
- byte peek (int addr)
- void poke (int addr, byte b)
- void _set (int addr, byte b)
- const char ∗ getName ()
- int getSize ()

Additional Inherited Members

4.9.1 Constructor & Destructor Documentation

4.9.1.1 MemoryDevice::MemoryDevice (int *size_in_bytes*, short int *ro*)

4.9.1.2 MemoryDevice::∼MemoryDevice ( )

4.9.2 Member Function Documentation

4.9.2.1 void MemoryDevice::_set (int *addr*, byte *b*)

4.9.2.2 const char∗ MemoryDevice::getName ( )

Reimplemented from Device.

4.9.2.3 int MemoryDevice::getSize ( )

Reimplemented from Device.

4.9.2.4 byte MemoryDevice::peek (int *addr*)

Reimplemented from Device.

4.9.2.5 void MemoryDevice::poke (int *addr*, byte *b*)

Reimplemented from Device.

The documentation for this class was generated from the following files:

- MemoryDevice.h
- MemoryDevice.cpp

4.10 MemoryMap Class Reference

```
#include <MemoryMap.h>
```

Classes

- struct **Node**

Public Member Functions

- MemoryMap (int size)
- ∼MemoryMap ()
- bool connect (Device ∗dev, int addr)
- byte peek (int addr)
- void poke (int addr, byte b)
- word peekw (int addr)
- void pokew (int addr, word w)
- void add (Device ∗dev, int from, int to)
- void dump ()
- bool getAddressName (char ∗str, int len, int addr)

4.10.1 Constructor & Destructor Documentation

4.10.1.1 MemoryMap::MemoryMap (int *size*)

4.10.1.2 MemoryMap::∼MemoryMap ( )

4.10.2 Member Function Documentation

4.10.2.1 void MemoryMap::add (Device ∗ *dev,* int *from,* int *to*)

4.10.2.2 bool MemoryMap::connect (Device ∗ *dev,* int *addr*)

4.10.2.3 void MemoryMap::dump ( )

4.10.2.4 bool MemoryMap::getAddressName (char ∗ *str,* int *len,* int *addr*)

4.10.2.5 byte MemoryMap::peek (int *addr*)

4.10.2.6 word MemoryMap::peekw (int *addr*)

4.10.2.7 void MemoryMap::poke (int *addr,* byte *b*)

4.10.2.8 void MemoryMap::pokew (int *addr,* word *w*)

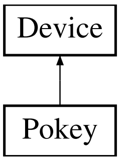
The documentation for this class was generated from the following files:

- MemoryMap.h
- MemoryMap.cpp

4.11 Pokey Class Reference

```
#include <Pokey.h>
```

Inheritance diagram for Pokey:



Public Member Functions

- Pokey ()
- ∼Pokey ()
- byte peek (int addr)
- void poke (int addr, byte b)
- const char ∗ getName ()
- int getSize ()
- bool getAddressName (char ∗str, int len, int addr)

Additional Inherited Members

4.11.1 Constructor & Destructor Documentation

4.11.1.1 Pokey::Pokey ( )

4.11.1.2 Pokey::∼Pokey ( )

4.11.2 Member Function Documentation

4.11.2.1 bool Pokey::getAddressName (char ∗ *str,* int *len,* int *addr*) [virtual]

Reimplemented from Device.

4.11.2.2 const char∗ Pokey::getName ( )

Reimplemented from Device.

4.11.2.3 int Pokey::getSize ( )

Reimplemented from Device.

4.11.2.4 byte Pokey::peek (int *addr*)

Reimplemented from Device.

4.11.2.5 void Pokey::poke (int *addr,* byte *b*)

Reimplemented from Device.

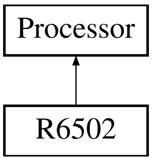
The documentation for this class was generated from the following files:

- Pokey.h
- Pokey.cpp

4.12 Processor Class Reference

```
#include <Processor.h>
```

Inheritance diagram for Processor:



Public Member Functions

- Processor (MemoryMap ∗mem)
- ∼Processor ()
- virtual void reset ()

- virtual void step ()
- virtual void run ()
- virtual MemoryMap ∗ getMemory ()
- virtual int disassemble (int addr, char ∗str, int len)

Protected Attributes

- MemoryMap ∗ memory
- bool running

4.12.1 Constructor & Destructor Documentation

4.12.1.1 Processor::Processor (**MemoryMap** ∗ *mem*)

4.12.1.2 Processor::∼Processor ( )

4.12.2 Member Function Documentation

4.12.2.1 int Processor::disassemble (int *addr,* char ∗ *str,* int *len*) [virtual]

4.12.2.2 **MemoryMap** ∗ Processor::getMemory ( ) [virtual]

Reimplemented in R6502.

4.12.2.3 void Processor::reset ( ) [virtual]

Reimplemented in R6502.

4.12.2.4 void Processor::run ( ) [virtual]

Reimplemented in R6502.

4.12.2.5 void Processor::step ( ) [virtual]

Reimplemented in R6502.

4.12.3 Member Data Documentation

4.12.3.1 **MemoryMap**∗ Processor::memory [protected]

4.12.3.2 bool Processor::running [protected]

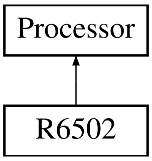
The documentation for this class was generated from the following files:

- Processor.h
- Processor.cpp

4.13 R6502 Class Reference

```
#include <R6502.h>
```

Inheritance diagram for R6502:



Public Member Functions

- R6502 (MemoryMap ∗mem)
- ∼R6502 ()
- void reset ()
- void step ()
- void run ()
- void summary ()
- R6502State ∗ getState ()
- MemoryMap ∗ getMemory ()

Protected Attributes

- R6502State ∗ state
- int irAddressMode
- int irOpMode
- int effectiveAddress
- int lo
- int hi
- int tmp
- long opBeginTime
- int behaviour

4.13.1 Constructor & Destructor Documentation

4.13.1.1 R6502::R6502 (**MemoryMap** ∗ *mem*)

4.13.1.2 R6502::∼R6502 ( )

4.13.2 Member Function Documentation

4.13.2.1 **MemoryMap** ∗ R6502::getMemory ( ) `[virtual]`

Reimplemented from Processor.

4.13.2.2 R6502State ∗ R6502::getState ( )

4.13.2.3 void R6502::reset ( ) `[virtual]`

Reimplemented from Processor.

4.13.2.4 void R6502::run ( ) [virtual]

Reimplemented from Processor.

4.13.2.5 void R6502::step ( ) [virtual]

Single Byte Instructions; Implied and Relative

JMP

ORA - Logical Inclusive Or

ASL - Arithmetic Shift Left

BIT - Bit Test

AND - Logical AND

ROL - Rotate Left

EOR - Exclusive OR

LSR - Logical Shift Right

ADC - Add with Carry

ROR - Rotate Right

STA - Store Accumulator

STY - Store Y Register

STX - Store X Register

LDY - Load Y Register

LDX - Load X Register

LDA - Load Accumulator

CPY - Compare Y Register

CMP - Compare Accumulator

DEC - Decrement Memory

CPX - Compare X Register

SBC - Subtract with Carry (Borrow)

INC - Increment Memory

Unimplemented Instructions

Reimplemented from Processor.

4.13.2.6 void R6502::summary ( )

4.13.3 Member Data Documentation

4.13.3.1 int R6502::behaviour [protected]

4.13.3.2 int R6502::effectiveAddress [protected]

4.13.3.3 int R6502::hi [protected]

4.13.3.4 int R6502::irAddressMode [protected]

4.13.3.5 int R6502::irOpMode [protected]

4.13.3.6 int R6502::lo [protected]

4.13.3.7 long R6502::opBeginTime [protected]

4.13.3.8 R6502State* R6502::state [protected]

4.13.3.9 int R6502::tmp [protected]

The documentation for this class was generated from the following files:

- R6502.h
- R6502.cpp

4.14 Timer Class Reference

```
#include <Timer.h>
```

Public Member Functions

- Timer ()
- ~Timer ()

Static Public Member Functions

- static void timerInterruptHandler (int sig)
- static void addListener (TimerListener *lstnr, useconds_t usec)
- static void start ()
- static void stop ()

4.14.1 Constructor & Destructor Documentation

4.14.1.1 Timer::Timer ( )

4.14.1.2 Timer::~Timer ( )

4.14.2 Member Function Documentation

4.14.2.1 void Timer::addListener (TimerListener * *lstnr,* useconds_t *usec*) [static]

4.14.2.2 void Timer::start ( ) [static]

4.14.2.3 void Timer::stop ( ) [static]

4.14.2.4 void Timer::timerInterruptHandler (int *sig*) [static]

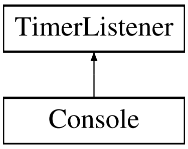
The documentation for this class was generated from the following files:

- Timer.h
- Timer.cpp

4.15 TimerListener Class Reference

```
#include <TimerListener.h>
```

Inheritance diagram for TimerListener:



Public Member Functions

- virtual void timerFired ()

4.15.1 Member Function Documentation

4.15.1.1 void TimerListener::timerFired ( ) [virtual]

Reimplemented in Console.

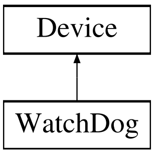
The documentation for this class was generated from the following files:

- TimerListener.h
- TimerListener.cpp

4.16 WatchDog Class Reference

```
#include <WatchDog.h>
```

Inheritance diagram for WatchDog:



Public Member Functions

- WatchDog ()
- ∼WatchDog ()
- byte peek (int addr)
- void poke (int addr, byte b)
- const char ∗ getName ()
- int getSize ()
- bool getAddressName (char ∗str, int len, int addr)

Additional Inherited Members

4.16.1 Constructor & Destructor Documentation

4.16.1.1 WatchDog::WatchDog ( )

4.16.1.2 WatchDog::∼WatchDog ( )

4.16.2 Member Function Documentation

4.16.2.1 bool WatchDog::getAddressName (char ∗ *str,* int *len,* int *addr*) [virtual]

Reimplemented from Device.

4.16.2.2 const char∗ WatchDog::getName ( )

Reimplemented from Device.

4.16.2.3 int WatchDog::getSize ( )

Reimplemented from Device.

4.16.2.4 byte WatchDog::peek (int *addr*)

Reimplemented from Device.

4.16.2.5 void WatchDog::poke (int *addr,* byte *b*)

Reimplemented from Device.

The documentation for this class was generated from the following files:

- WatchDog.h
- WatchDog.cpp

Chapter 5

File Documentation

5.1 ADInput.cpp File Reference

```
#include <stdio.h>
#include <string.h>
#include "ADInput.h"
```

5.2 ADInput.h File Reference

```
#include "Device.h"
```

Classes

- class ADInput

5.3 ADMachine.cpp File Reference

```
#include "AsteroidsDeluxe.h"
```

5.4 ADOutput.cpp File Reference

```
#include <stdio.h>
#include <string.h>
#include "ADOutput.h"
```

5.5 ADOutput.h File Reference

```
#include "Device.h"
```

Classes

- class ADOutput

5.6 CodeWindow.cpp File Reference

```
#include <ncurses.h>
#include "CodeWindow.h"
```

5.7 CodeWindow.h File Reference

```
#include <ncurses.h>
#include "R6502.h"
#include "MemoryMap.h"
#include "CPUState.h"
#include "Window.h"
```

Classes

- class CodeWindow

5.8 Console.cpp File Reference

```
#include <ncurses.h>
#include "gem.h"
#include "Console.h"
#include "Timer.h"
```

5.9 Console.h File Reference

```
#include <ncurses.h>
#include "R6502.h"
#include "CodeWindow.h"
#include "TimerListener.h"
```

Classes

- class Console

5.10 CPUState.cpp File Reference

```
#include <stdio.h>
```

```
#include <string.h>
#include "R6502.h"
#include "CPUState.h"
#include "Instructions.h"
```

5.11 CPUState.h File Reference

```
#include <ctype.h>
#include "MemoryMap.h"
```

Classes

- class CPUState

5.12 Device.cpp File Reference

```
#include <string.h>
#include "gem.h"
#include "Device.h"
```

5.13 Device.h File Reference

```
#include "gem.h"
#include "DeviceListener.h"
```

Classes

- class Device

Macros

- #define MAXDEVNAME 128

5.13.1 Macro Definition Documentation

5.13.1.1 #define MAXDEVNAME 128

5.14 DeviceListener.cpp File Reference

```
#include "DeviceListener.h"
```

5.15 DeviceListener.h File Reference

```
#include "Device.h"
```

Classes

- class DeviceListener

Macros

- #define _DEVICE_LISTENER_H

5.15.1 Macro Definition Documentation

5.15.1.1 #define _DEVICE_LISTENER_H

5.16 Machine.cpp File Reference

```
#include "Machine.h"
```

5.17 Machine.h File Reference

```
#include "Processor.h"
#include "MemoryMap.h"
```

Classes

- class Machine

5.18 MemoryDevice.cpp File Reference

```
#include <exception>
#include <malloc.h>
#include "gem.h"
#include "MemoryDevice.h"
```

5.19 MemoryDevice.h File Reference

```
#include "Device.h"
```

Classes

- class MemoryDevice

5.20 MemoryMap.cpp File Reference

```
#include <stdio.h>
#include "MemoryMap.h"
```

5.21 MemoryMap.h File Reference

```
#include "Device.h"
```

Classes

- class MemoryMap
- struct **MemoryMap::Node**

5.22 Pokey.cpp File Reference

```
#include <exception>
#include <string.h>
#include <malloc.h>
#include "gem.h"
#include "Pokey.h"
```

5.23 Pokey.h File Reference

```
#include "Device.h"
```

Classes

- class Pokey

Macros

- #define POKEY_SIZE 16

5.23.1 Macro Definition Documentation

5.23.1.1 #define POKEY_SIZE 16

5.24 Processor.cpp File Reference

```
#include <string.h>
#include "Processor.h"
```

5.25 Processor.h File Reference

```
#include "MemoryMap.h"
```

Classes

- class Processor

5.26 R6502.cpp File Reference

```
#include <stdio.h>
#include <time.h>
#include "R6502.h"
```

5.27 R6502.h File Reference

```
#include "gem.h"
#include "Processor.h"
#include "MemoryMap.h"
#include "R6502State.h"
#include "R6502Instructions.h"
```

Classes

- class R6502

Macros

- #define P_CARRY 0x01
- #define P_ZERO 0x02
- #define P_IRQ_DISABLE 0x04
- #define P_DECIMAL 0x08
- #define P_BREAK 0x10
- #define P_OVERFLOW 0x40
- #define P_NEGATIVE 0x80
- #define NMI_VECTOR_L 0xfffa
- #define NMI_VECTOR_H 0xfffb
- #define RST_VECTOR_L 0xfffc
- #define RST_VECTOR_H 0xfffd
- #define IRQ_VECTOR_L 0xfffe
- #define IRQ_VECTOR_H 0xffff
- #define NMOS_WITH_INDIRECT_JMP_BUG 1
- #define NMOS_WITH_ROR_BUG 2
- #define MODE_NUL 0
- #define MODE_ACC 1
- #define MODE_ABS 2
- #define MODE_ABX 3
- #define MODE_ABY 4

- #define MODE_IMM 5
- #define MODE_IMP 6
- #define MODE_IND 7
- #define MODE_INX 8
- #define MODE_INY 9
- #define MODE_REL 10
- #define MODE_ZPG 11
- #define MODE_ZPX 12

5.27.1 Macro Definition Documentation

5.27.1.1 #define IRQ_VECTOR_H 0xffff

5.27.1.2 #define IRQ_VECTOR_L 0xfffe

5.27.1.3 #define MODE_ABS 2

5.27.1.4 #define MODE_ABX 3

5.27.1.5 #define MODE_ABY 4

5.27.1.6 #define MODE_ACC 1

5.27.1.7 #define MODE_IMM 5

5.27.1.8 #define MODE_IMP 6

5.27.1.9 #define MODE_IND 7

5.27.1.10 #define MODE_INX 8

5.27.1.11 #define MODE_INY 9

5.27.1.12 #define MODE_NUL 0

5.27.1.13 #define MODE_REL 10

5.27.1.14 #define MODE_ZPG 11

5.27.1.15 #define MODE_ZPX 12

5.27.1.16 #define NMI_VECTOR_H 0xfffb

5.27.1.17 #define NMI_VECTOR_L 0xfffa

5.27.1.18 #define NMOS_WITH_INDIRECT_JMP_BUG 1

5.27.1.19 #define NMOS_WITH_ROR_BUG 2

5.27.1.20 #define P_BREAK 0x10

5.27.1.21 #define P_CARRY 0x01

5.27.1.22 #define P_DECIMAL 0x08

5.27.1.23 #define P_IRQ_DISABLE 0x04

5.27.1.24 #define P_NEGATIVE 0x80

5.27.1.25 #define P_OVERFLOW 0x40

5.27.1.26 #define P_ZERO 0x02

5.27.1.27 #define RST_VECTOR_H 0xfffd

5.27.1.28 #define RST_VECTOR_L 0xfffc

5.28 Timer.cpp File Reference

```
#include <stdio.h>
#include <unistd.h>
#include <signal.h>
#include <ncurses.h>
#include "Timer.h"
#include "Window.h"
```

5.29 Timer.h File Reference

```
#include "TimerListener.h"
#include "Window.h"
```

Classes

- class Timer

Macros

- #define MAX_TIMER_LISTENERS 10
- #define MAX_TIMER_JITTER 4

5.29.1 Macro Definition Documentation

5.29.1.1 #define MAX_TIMER_JITTER 4

5.29.1.2 #define MAX_TIMER_LISTENERS 10

5.30 TimerListener.cpp File Reference

```
#include "TimerListener.h"
```

5.31 TimerListener.h File Reference

Classes

- class TimerListener

5.32 WatchDog.cpp File Reference

```
#include <exception>
#include <string.h>
#include <malloc.h>
#include "AtariEm.h"
#include "WatchDog.h"
```

5.33 WatchDog.h File Reference

```
#include "Device.h"
```

Classes

- class WatchDog

Index