// ==============================================================

// RTL generated by Vivado(TM) HLS - High-Level Synthesis from C, C++ and SystemC

// Version: 2017.2

// Copyright (C) 1986-2017 Xilinx, Inc. All Rights Reserved.

//

// ===========================================================

#ifndef \_simple\_HH\_

#define \_simple\_HH\_

#include "systemc.h"

#include "AESL\_pkg.h"

#include "simple\_fadd\_32ns\_bkb.h"

namespace ap\_rtl {

struct simple : public sc\_module {

// Port declarations 39

sc\_in\_clk ap\_clk;

sc\_in< sc\_logic > ap\_rst;

sc\_in< sc\_logic > ap\_start;

sc\_out< sc\_logic > ap\_done;

sc\_out< sc\_logic > ap\_idle;

sc\_out< sc\_logic > ap\_ready;

sc\_in< sc\_lv<32> > a\_0;

sc\_in< sc\_lv<32> > a\_1;

sc\_in< sc\_lv<32> > a\_2;

sc\_in< sc\_lv<32> > a\_3;

sc\_in< sc\_lv<32> > a\_4;

sc\_in< sc\_lv<32> > a\_5;

sc\_in< sc\_lv<32> > a\_6;

sc\_in< sc\_lv<32> > a\_7;

sc\_in< sc\_lv<32> > b\_0;

sc\_in< sc\_lv<32> > b\_1;

sc\_in< sc\_lv<32> > b\_2;

sc\_in< sc\_lv<32> > b\_3;

sc\_in< sc\_lv<32> > b\_4;

sc\_in< sc\_lv<32> > b\_5;

sc\_in< sc\_lv<32> > b\_6;

sc\_in< sc\_lv<32> > b\_7;

sc\_out< sc\_lv<32> > c\_0;

sc\_out< sc\_logic > c\_0\_ap\_vld;

sc\_out< sc\_lv<32> > c\_1;

sc\_out< sc\_logic > c\_1\_ap\_vld;

sc\_out< sc\_lv<32> > c\_2;

sc\_out< sc\_logic > c\_2\_ap\_vld;

sc\_out< sc\_lv<32> > c\_3;

sc\_out< sc\_logic > c\_3\_ap\_vld;

sc\_out< sc\_lv<32> > c\_4;

sc\_out< sc\_logic > c\_4\_ap\_vld;

sc\_out< sc\_lv<32> > c\_5;

sc\_out< sc\_logic > c\_5\_ap\_vld;

sc\_out< sc\_lv<32> > c\_6;

sc\_out< sc\_logic > c\_6\_ap\_vld;

sc\_out< sc\_lv<32> > c\_7;

sc\_out< sc\_logic > c\_7\_ap\_vld;

sc\_out< sc\_lv<32> > ap\_return;

sc\_signal< sc\_logic > ap\_var\_for\_const0;

// Module declarations

simple(sc\_module\_name name);

SC\_HAS\_PROCESS(simple);

~simple();

sc\_trace\_file\* mVcdFile;

ofstream mHdltvinHandle;

ofstream mHdltvoutHandle;

simple\_fadd\_32ns\_bkb<1,4,32,32,32>\* simple\_fadd\_32ns\_bkb\_U1;

simple\_fadd\_32ns\_bkb<1,4,32,32,32>\* simple\_fadd\_32ns\_bkb\_U2;

simple\_fadd\_32ns\_bkb<1,4,32,32,32>\* simple\_fadd\_32ns\_bkb\_U3;

simple\_fadd\_32ns\_bkb<1,4,32,32,32>\* simple\_fadd\_32ns\_bkb\_U4;

simple\_fadd\_32ns\_bkb<1,4,32,32,32>\* simple\_fadd\_32ns\_bkb\_U5;

simple\_fadd\_32ns\_bkb<1,4,32,32,32>\* simple\_fadd\_32ns\_bkb\_U6;

simple\_fadd\_32ns\_bkb<1,4,32,32,32>\* simple\_fadd\_32ns\_bkb\_U7;

simple\_fadd\_32ns\_bkb<1,4,32,32,32>\* simple\_fadd\_32ns\_bkb\_U8;

sc\_signal< sc\_lv<4> > ap\_CS\_fsm;

sc\_signal< sc\_logic > ap\_CS\_fsm\_state1;

sc\_signal< sc\_lv<32> > grp\_fu\_212\_p2;

sc\_signal< sc\_logic > ap\_CS\_fsm\_state4;

sc\_signal< sc\_lv<32> > grp\_fu\_219\_p2;

sc\_signal< sc\_lv<32> > grp\_fu\_226\_p2;

sc\_signal< sc\_lv<32> > grp\_fu\_233\_p2;

sc\_signal< sc\_lv<32> > grp\_fu\_240\_p2;

sc\_signal< sc\_lv<32> > grp\_fu\_247\_p2;

sc\_signal< sc\_lv<32> > grp\_fu\_254\_p2;

sc\_signal< sc\_lv<32> > grp\_fu\_261\_p2;

sc\_signal< sc\_lv<4> > ap\_NS\_fsm;

static const sc\_logic ap\_const\_logic\_1;

static const sc\_logic ap\_const\_logic\_0;

static const sc\_lv<4> ap\_ST\_fsm\_state1;

static const sc\_lv<4> ap\_ST\_fsm\_state2;

static const sc\_lv<4> ap\_ST\_fsm\_state3;

static const sc\_lv<4> ap\_ST\_fsm\_state4;

static const sc\_lv<32> ap\_const\_lv32\_0;

static const sc\_lv<32> ap\_const\_lv32\_3;

static const bool ap\_const\_boolean\_1;

// Thread declarations

void thread\_ap\_var\_for\_const0();

void thread\_ap\_clk\_no\_reset\_();

void thread\_ap\_CS\_fsm\_state1();

void thread\_ap\_CS\_fsm\_state4();

void thread\_ap\_done();

void thread\_ap\_idle();

void thread\_ap\_ready();

void thread\_ap\_return();

void thread\_c\_0();

void thread\_c\_0\_ap\_vld();

void thread\_c\_1();

void thread\_c\_1\_ap\_vld();

void thread\_c\_2();

void thread\_c\_2\_ap\_vld();

void thread\_c\_3();

void thread\_c\_3\_ap\_vld();

void thread\_c\_4();

void thread\_c\_4\_ap\_vld();

void thread\_c\_5();

void thread\_c\_5\_ap\_vld();

void thread\_c\_6();

void thread\_c\_6\_ap\_vld();

void thread\_c\_7();

void thread\_c\_7\_ap\_vld();

void thread\_ap\_NS\_fsm();

void thread\_hdltv\_gen();

};

}

using namespace ap\_rtl;

#endif