```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

/* ASSERTIONS */
#define AssertionFlag 1

/* USER DEFINED TYPE DATA */
#include "abro.h"
extern int _check_data(char *);
extern char *_data_to_text(data);
extern void _text_to_data(data *, char *);
extern int _eq_data(data,data);
extern int _ne_data(data,data);

static data __text_to_data( char *st) {
    data back;
    _text_to_data(&back, st);
    return back;
}


extern int _less_data(data,data);
static int _great_data(data av1,data av2) {
    return _less_data(av2, av1);
}
static int _lesseq_data(data av1,data av2) {
    return _less_data(av1, av2) || _eq_data(av1, av2);
}
static int _greateq_data(data av1,data av2) {
    return _lesseq_data(av2, av1);
}


static char *_strcat(char *av1, char *av2) {
    char *back;
    int size_av1 = !av1 ? 0 : strlen(av1);
    int size_av2 = !av2 ? 0 : strlen(av2);
    if (!size_av1) return av2;
    if (!size_av2) return av1;
    back = (char *)malloc(size_av1 + size_av2 + 1);
    if (!back) return (char *)0;
    strcpy(back, av1);
    strcpy(back + size_av1, av2);
    back[size_av1 + size_av2] = '\0';
    return back;
}
static int _strcmp(char *av1, char *av2) {
    if (!av1 || !av2) return 0;
    return strcmp(av1, av2);
}

#define __strcat(Ls, Lv, Rs, Rv)    _strcat((Lv),(Rv))
#define __streq(Ls, Lv, Rs, Rv)     !_strcmp((Lv),(Rv))
#define __strneq(Ls, Lv, Rs, Rv)    _strcmp((Lv),(Rv))
#define __strless(Ls, Lv, Rs, Rv)   _strcmp((Lv),(Rv)) < 0
#define __strlesseq(Ls, Lv, Rs, Rv) _strcmp((Lv),(Rv)) <= 0
#define __strgreat(Ls, Lv, Rs, Rv)  _strcmp((Lv),(Rv)) > 0
#define __strgreateq(Ls, Lv, Rs, Rv) _strcmp((Lv),(Rv)) >= 0
```

```c
#define __eq_data(Ls, Lv, Rs, Rv)      _eq_data((Lv), (Rv))
#define __neq_data(Ls, Lv, Rs, Rv)     _ne_data((Lv), (Rv))
#define __less_data(Ls, Lv, Rs, Rv)    _less_data((Lv), (Rv))
#define __lesseq_data(Ls, Lv, Rs, Rv)  _lesseq_data((Lv),(Rv))
#define __great_data(Ls, Lv, Rs, Rv)   _great_data((Lv),(Rv))
#define __greateq_data(Ls, Lv, Rs, Rv) _greateq_data((Lv),(Rv))
/* ABRO ACTIONS */
extern int ABRO_run();
extern int ABRO_reset();

/* INPUTS */
static int ABRO_A_wrapper_status = 0;
static char * ABRO_A_wrapper_value;
extern void ABRO_I_A(char *);

static int ABRO_B_wrapper_status = 0;
static data ABRO_B_wrapper_value;
extern void ABRO_I_B(data);

static int ABRO_R_wrapper_status = 0;
extern void ABRO_I_R(void);


/* OUTPUTS */
static int ABRO_O_wrapper_status = 0;
static data ABRO_O_wrapper_value;
void ABRO_O_O(data _value) {
    ABRO_O_wrapper_status = 1;
    ABRO_O_wrapper_value = _value;
}


/* SENSORS */

/* SIGNAL INSTRUMENTATION */
#ifdef STATE_DUMP
    extern void ABRO_state_dump_init();
    extern void ABRO_state_dump_end();
#endif /* STATE_DUMP */
#ifdef SIGNAL_RECORD
    extern void ABRO_signal_record_init(char *);
    extern void ABRO_signal_record_end();
    extern void ABRO_signal_record_comment(char *);
#endif /* SIGNAL_RECORD */


/* MAIN */
int main(int ac, char *av[]) {
#ifdef STATE_DUMP
    ABRO_state_dump_init("dump_abro.blif");
#endif /* STATE_DUMP */
#ifdef SIGNAL_RECORD
    ABRO_signal_record_init("record_abro.eso");
#endif /* SIGNAL_RECORD */

    ABRO_reset();
    ABRO_A_wrapper_status = 1;
    ABRO_I_A("1");
```

```c
    ABRO_A_wrapper_value = "1";
    ABRO_B_wrapper_status = 0;
    ABRO_R_wrapper_status = 0;
    ABRO_O_wrapper_status = 0;
#ifdef SIGNAL_RECORD
    ABRO_signal_record_comment("Cycle 1");
#endif /* SIGNAL_RECORD */
    /* Sync on CLK rising edge */
    ABRO_run();

    /* Outputs checking */


/**********************************/
/* CLK cycle number: 2 */
/**********************************/
    /* Inputs initialization */
    ABRO_B_wrapper_status = 1;
    if (!_check_data("premier tirage")) {
#ifdef STATE_DUMP
        ABRO_state_dump_end();
#endif /* STATE_DUMP */
#ifdef SIGNAL_RECORD
        ABRO_signal_record_comment("Run-time checking type error, file abro.esi, line 3: valued signal B o
f type \"data\" set on \"premier tirage\".");
        ABRO_signal_record_end();
#endif /* SIGNAL_RECORD */
        return 1;
    }
    ABRO_I_B(__text_to_data("premier tirage"));
    ABRO_B_wrapper_value = __text_to_data("premier tirage");
    ABRO_A_wrapper_status = 0;
    ABRO_R_wrapper_status = 0;
    ABRO_O_wrapper_status = 0;
#ifdef SIGNAL_RECORD
    ABRO_signal_record_comment("Cycle 2");
#endif /* SIGNAL_RECORD */
    /* Sync on CLK rising edge */
    ABRO_run();

    /* Outputs checking */


/**********************************/
/* CLK cycle number: 3 */
/**********************************/
    /* Inputs initialization */
    ABRO_A_wrapper_status = 1;
    ABRO_I_A("2");
    ABRO_A_wrapper_value = "2";
    ABRO_B_wrapper_status = 0;
    ABRO_R_wrapper_status = 0;
    ABRO_O_wrapper_status = 0;
#ifdef SIGNAL_RECORD
    ABRO_signal_record_comment("Cycle 3");
#endif /* SIGNAL_RECORD */
    /* Sync on CLK rising edge */
    ABRO_run();
```

```c
    /* Outputs checking */


/**********************************/
/* CLK cycle number: 4 */
/**********************************/
    /* Inputs initialization */
    ABRO_B_wrapper_status = 1;
    if (!_check_data("second tirage")) {
#ifdef STATE_DUMP
        ABRO_state_dump_end();
#endif /* STATE_DUMP */
#ifdef SIGNAL_RECORD
        ABRO_signal_record_comment("Run-time checking type error, file abro.esi, line 9: valued signal B o
f type \"data\" set on \"second tirage\".");
        ABRO_signal_record_end();
#endif /* SIGNAL_RECORD */
        return 1;
    }
    ABRO_I_B(__text_to_data("second tirage"));
    ABRO_B_wrapper_value = __text_to_data("second tirage");
    ABRO_A_wrapper_status = 0;
    ABRO_R_wrapper_status = 0;
    ABRO_O_wrapper_status = 0;
#ifdef SIGNAL_RECORD
    ABRO_signal_record_comment("Cycle 4");
#endif /* SIGNAL_RECORD */
    /* Sync on CLK rising edge */
    ABRO_run();

    /* Outputs checking */


/**********************************/
/* CLK cycle number: 5 */
/**********************************/
    /* Inputs initialization */
    if (AssertionFlag) {
        if (!(__streq(ABROA_wrapper_status, ABRO_A_wrapper_value, 2, "1") && ABRO_
B_wrapper_status)) {
#ifdef SIGNAL_RECORD
            ABRO_signal_record_comment("%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%");
            ABRO_signal_record_comment("NOTE: Break point reached");
            ABRO_signal_record_comment(" file 'abro.esi'");
            ABRO_signal_record_comment(" line '8'");
            ABRO_signal_record_comment("%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%\n");
#endif /* SIGNAL_RECORD */
        }
    }
    ABRO_I_R();
    ABRO_R_wrapper_status = 1;
    ABRO_A_wrapper_status = 0;
    ABRO_B_wrapper_status = 0;
    ABRO_O_wrapper_status = 0;
#ifdef SIGNAL_RECORD
    ABRO_signal_record_comment("Cycle 5");
#endif /* SIGNAL_RECORD */
```

```
    /* Sync on CLK rising edge */
    ABRO_run();

    /* Outputs checking */


/**********************************/
/* CLK cycle number: 6 */
/**********************************/
    /* Inputs initialization */
    /* A (\"3\"); */
#ifdef SIGNAL_RECORD
    ABRO_signal_record_comment("A (\"3\");");
#endif /* SIGNAL_RECORD */
    /* B (\"troisieme tirage\"); */
#ifdef SIGNAL_RECORD
    ABRO_signal_record_comment("B (\"troisieme tirage\");");
#endif /* SIGNAL_RECORD */

#ifdef STATE_DUMP
    ABRO_state_dump_end();
#endif /* STATE_DUMP */
#ifdef SIGNAL_RECORD
    ABRO_signal_record_end();
#endif /* SIGNAL_RECORD */
    return 0;
}
/* END */
```