147:AS
last_byte = (size_in == BYTE) && (state_byte == BYTE_0) || (size_in == HALF_WORD) && (state_byte == BYTE_1) || (size_in == WORD) && (state_byte == BYTE_3);
152:AL
153:BL
next_state_full = state_full;
155:CS
state_full
state_full
state_full
state_full
156:CA
175:CA
157:CA
170:CA
176:BL
158:BL
171:BL
156:BS
next_state_full = (write)? WRITE_1 : EMPTY;
159:IF
177:IF
172:IF
!(last_byte)
last_byte
last_byte
165:BL
160:BL
173:BS
next_state_full = (write)? BYPASS : WRITE_1;
(last_byte && !write)
166:IF
161:IF
write
(!write)
178:BS
next_state_full = WRITE_1;
167:BS
next_state_full = WRITE_2;
162:BS
next_state_full = EMPTY;
Leaf_152:AL
138:AL
139:BL
140:IF
(!rst_n)
!((!rst_n))
141:NS
state_full <= EMPTY;
143:NS
state_full <= next_state_full;
Leaf_138:AL
192:AS
bypass_byte0 = state_full != BYPASS;
195:AS
has_data = (state_full == WRITE_2) || (state_full == BYPASS);
184:AS
buffer_full = (state_full == WRITE_2) && !last_byte || (state_full == BYPASS) && !last_byte;
209:AL
210:BL
next_state_byte = state_byte;
212:CS
state_byte
state_byte
state_byte
state_byte
state_byte
214:CA
226:CA
240:CA
213:CA
244:CA
215:BL
227:BL
213:BS
next_state_byte = (write)? BYTE_0 : IDLE;
216:IF
228:IF
(size_in == BYTE)
!((size_in == BYTE))
(size_in == HALF_WORD)
241:BL
next_state_byte = BYTE_3;
!((size_in == HALF_WORD))
217:BL
222:BL
next_state_byte = BYTE_1;
229:BL
245:BL
218:IF
230:IF
246:IF
236:BL
next_state_byte = BYTE_2;
(!write && !has_data)
!((has_data || write && !buffer_full))
(has_data || write && !buffer_full)
!((has_data || write && !buffer_full))
(has_data || write && !buffer_full)
219:BS
next_state_byte = IDLE;
233:BS
next_state_byte = IDLE;
231:BS
next_state_byte = BYTE_0;
249:BS
next_state_byte = IDLE;
247:BS
next_state_byte = BYTE_0;
Leaf_209:AL
201:AL
202:BL
203:IF
(!rst_n)
!((!rst_n))
204:NS
state_byte <= IDLE;
206:NS
state_byte <= next_state_byte;
Leaf_201:AL
276:AS
byte_en = (state_byte == BYTE_0) && ((size_in == HALF_WORD) || (size_in == WORD)) && (state_full != BYPASS) || last_byte && has_data;
273:AS
crc_out_en = state_byte != IDLE;
333:AS
set_crc_init_sel = state_byte == BYTE_0;
255:AL
187:AS
read_wait = state_byte != IDLE;
268:AS
bypass_size = !((state_full != BYPASS) && (state_byte != BYTE_0) || (state_full == BYPASS));
256:BL
byte_sel = 2'b00;
258:CS
state_byte
state_byte
state_byte
state_byte
259:CA
261:CA
262:CA
260:CA
259:BS
byte_sel = BYTE_0;
261:BS
byte_sel = BYTE_2;
262:BS
byte_sel = BYTE_3;
260:BS
byte_sel = BYTE_1;
Leaf_255:AL
317:BS
next_state_reset = (last_byte)? RESET : RESET_2;
Leaf_289:AL
281:AL
282:BL
283:IF
!((!rst_n))
(!rst_n)
286:NS
state_reset <= next_state_reset;
284:NS
state_reset <= NO_RESET;
Leaf_281:AL
reset_chain
343:AS
reset_pending = state_reset != NO_RESET;
289:AL
337:AS
clear_crc_init_sel = (state_reset == NO_RESET) && last_byte && reset_chain || (state_byte == IDLE) && reset_chain || (state_reset == RESET) && last_byte || (state_reset == WRITE) && last_byte || (state_reset == RESET_2) && last_byte;
290:BL
next_state_reset = state_reset;
292:CS
state_reset
state_reset
state_reset
state_reset
state_reset
314:CA
300:CA
321:CA
307:CA
293:CA
315:BL
301:BL
322:BL
308:BL
294:BL
316:IF
302:IF
323:IF
309:IF
295:IF
297:IF
!(reset_chain)
last_byte
!(last_byte)
!(last_byte)
last_byte
last_byte
!(last_byte)
(reset_chain && !has_data && (state_byte != IDLE) && !last_byte || reset_chain && has_data && last_byte)
(reset_chain && has_data && !last_byte)
319:BS
next_state_reset = (last_byte)? NO_RESET : WRITE;
303:BS
next_state_reset = NO_RESET;
305:BS
next_state_reset = (write)? WRITE : RESET;
326:BS
next_state_reset = RESET_2;
324:BS
next_state_reset = (write)? WRITE : RESET;
310:BS
next_state_reset = (write)? WRITE : RESET;
312:BS
next_state_reset = WAIT;
296:BS
next_state_reset = RESET;
298:BS
next_state_reset = WAIT;