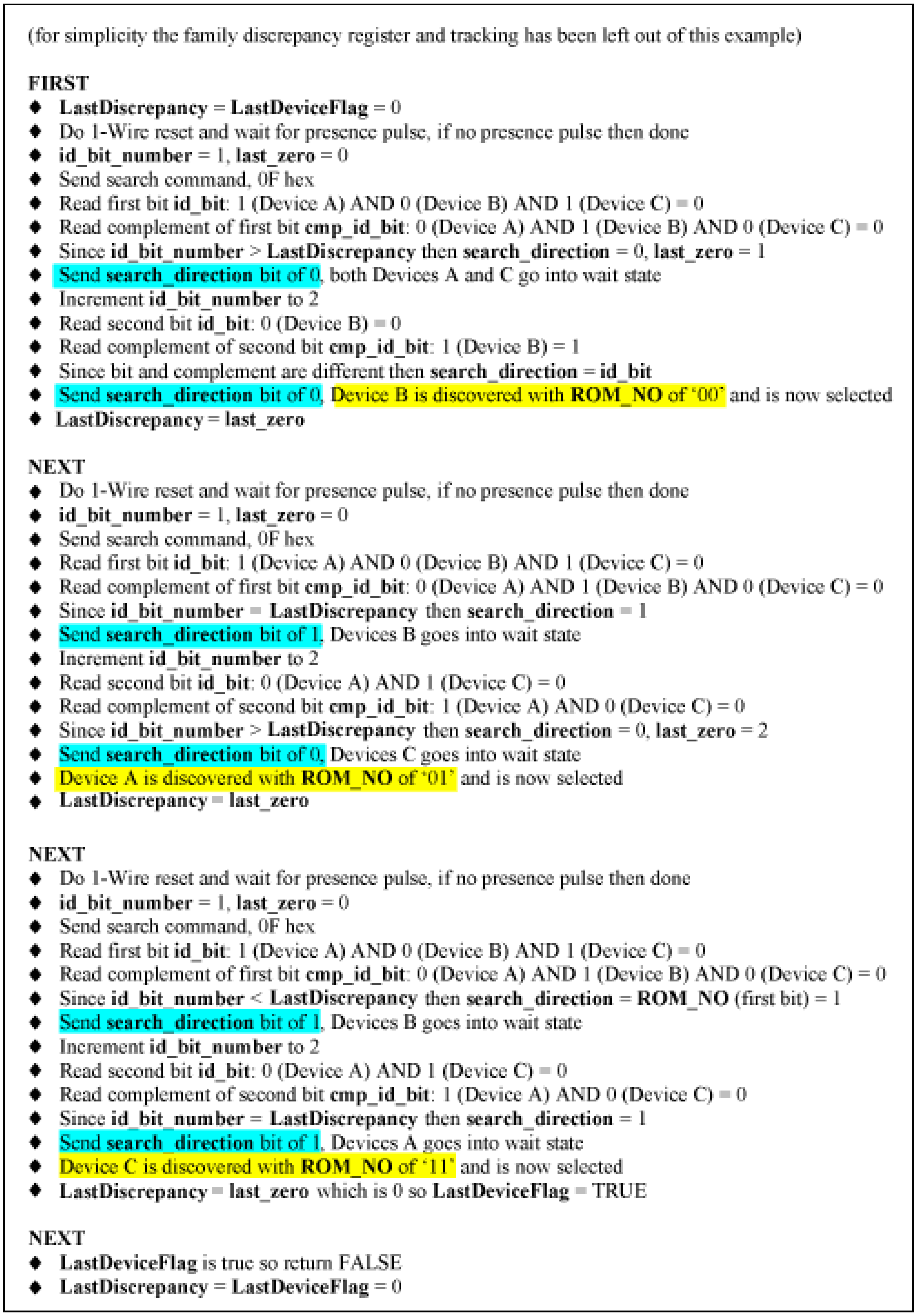

(for simplicity the family discrepancy register and tracking has been left out of this example)

FIRST

- **LastDiscrepancy** = **LastDeviceFlag** = 0
- Do 1-Wire reset and wait for presence pulse, if no presence pulse then done
- **id_bit_number** = 1, **last_zero** = 0
- Send search command, 0F hex
- Read first bit **id_bit**: 1 (Device A) AND 0 (Device B) AND 1 (Device C) = 0
- Read complement of first bit **cmp_id_bit**: 0 (Device A) AND 1 (Device B) AND 0 (Device C) = 0
- Since **id_bit_number** > **LastDiscrepancy** then **search_direction** = 0, **last_zero** = 1
- Send **search_direction** bit of 0, both Devices A and C go into wait state
- Increment **id_bit_number** to 2
- Read second bit **id_bit**: 0 (Device B) = 0
- Read complement of second bit **cmp_id_bit**: 1 (Device B) = 1
- Since bit and complement are different then **search_direction** = **id_bit**
- Send **search_direction** bit of 0, Device B is discovered with **ROM_NO** of '00' and is now selected
- **LastDiscrepancy** = **last_zero**

NEXT

- Do 1-Wire reset and wait for presence pulse, if no presence pulse then done
- **id_bit_number** = 1, **last_zero** = 0
- Send search command, 0F hex
- Read first bit **id_bit**: 1 (Device A) AND 0 (Device B) AND 1 (Device C) = 0
- Read complement of first bit **cmp_id_bit**: 0 (Device A) AND 1 (Device B) AND 0 (Device C) = 0
- Since **id_bit_number** = **LastDiscrepancy** then **search_direction** = 1
- Send **search_direction** bit of 1, Devices B goes into wait state
- Increment **id_bit_number** to 2
- Read second bit **id_bit**: 0 (Device A) AND 1 (Device C) = 0
- Read complement of second bit **cmp_id_bit**: 1 (Device A) AND 0 (Device C) = 0
- Since **id_bit_number** > **LastDiscrepancy** then **search_direction** = 0, **last_zero** = 2
- Send **search_direction** bit of 0, Devices C goes into wait state
- Device A is discovered with **ROM_NO** of '01' and is now selected
- **LastDiscrepancy** = **last_zero**

NEXT

- Do 1-Wire reset and wait for presence pulse, if no presence pulse then done
- **id_bit_number** = 1, **last_zero** = 0
- Send search command, 0F hex
- Read first bit **id_bit**: 1 (Device A) AND 0 (Device B) AND 1 (Device C) = 0
- Read complement of first bit **cmp_id_bit**: 0 (Device A) AND 1 (Device B) AND 0 (Device C) = 0
- Since **id_bit_number** < **LastDiscrepancy** then **search_direction** = **ROM_NO** (first bit) = 1
- Send **search_direction** bit of 1, Devices B goes into wait state
- Increment **id_bit_number** to 2
- Read second bit **id_bit**: 0 (Device A) AND 1 (Device C) = 0
- Read complement of second bit **cmp_id_bit**: 1 (Device A) AND 0 (Device C) = 0
- Since **id_bit_number** = **LastDiscrepancy** then **search_direction** = 1
- Send **search_direction** bit of 1, Devices A goes into wait state
- Device C is discovered with **ROM_NO** of '11' and is now selected
- **LastDiscrepancy** = **last_zero** which is 0 so **LastDeviceFlag** = TRUE

NEXT

- **LastDeviceFlag** is true so return FALSE
- **LastDiscrepancy** = **LastDeviceFlag** = 0

Figure 3. Search Example.

Advanced Search Variations

There are three advanced search variations using the same state information, namely LastDiscrepancy, LastFamilyDiscrepancy, LastDeviceFlag, and ROM_NO. These variations allow specific family types to be