**Digital Design and Computer Architecture LU**

# Lab Protocol

# Exercise II

Vorname Nachname, Matr. Nr. 0123456
e0123456@student.tuwien.ac.at

Vienna, May 12, 2022

# Task 1: VGA Graphics Controller

**Subtask 1**

VGA Oscilloscope Measurements

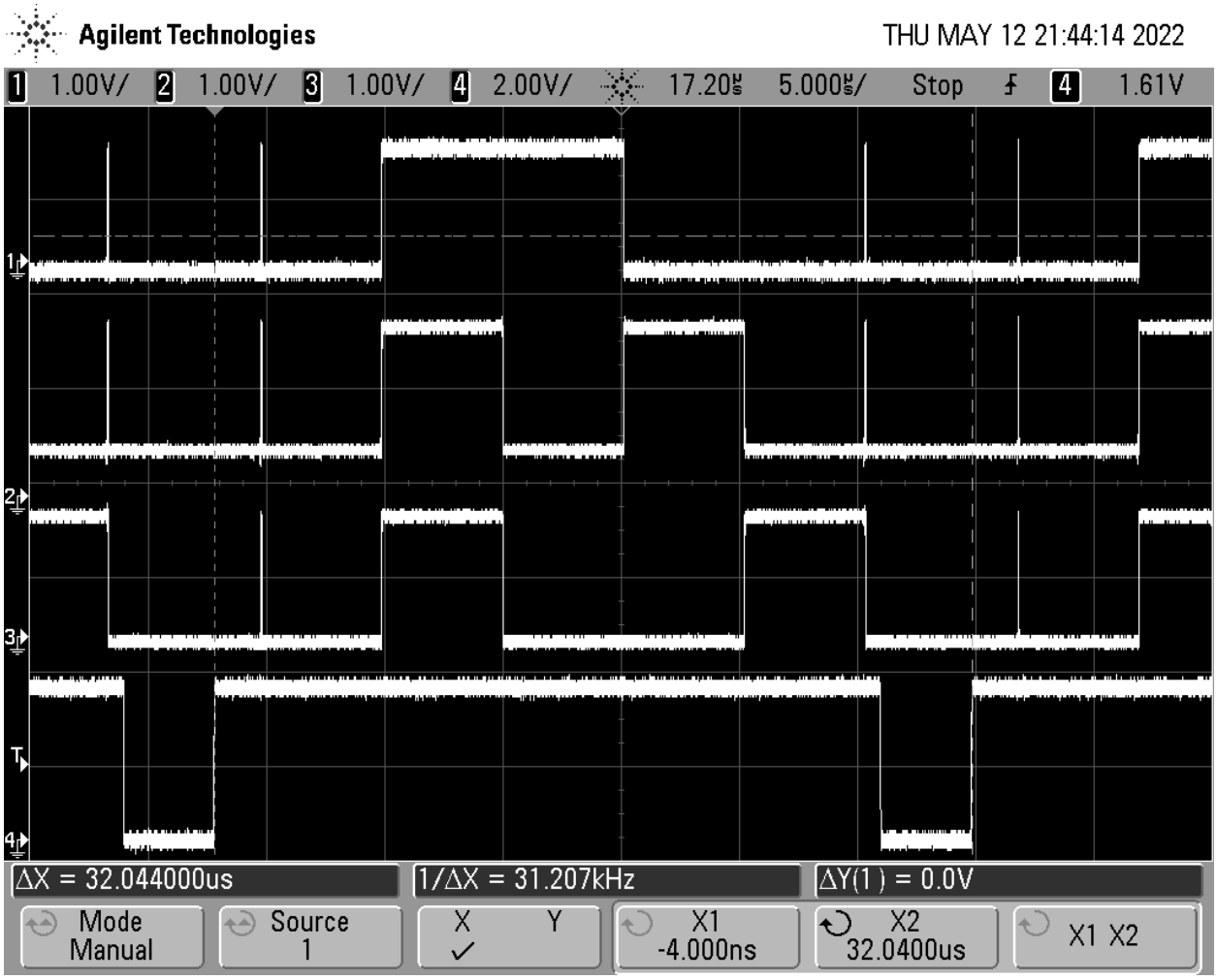


Figure 1: Line measurement with cursors marking the length (duration) of the whole line

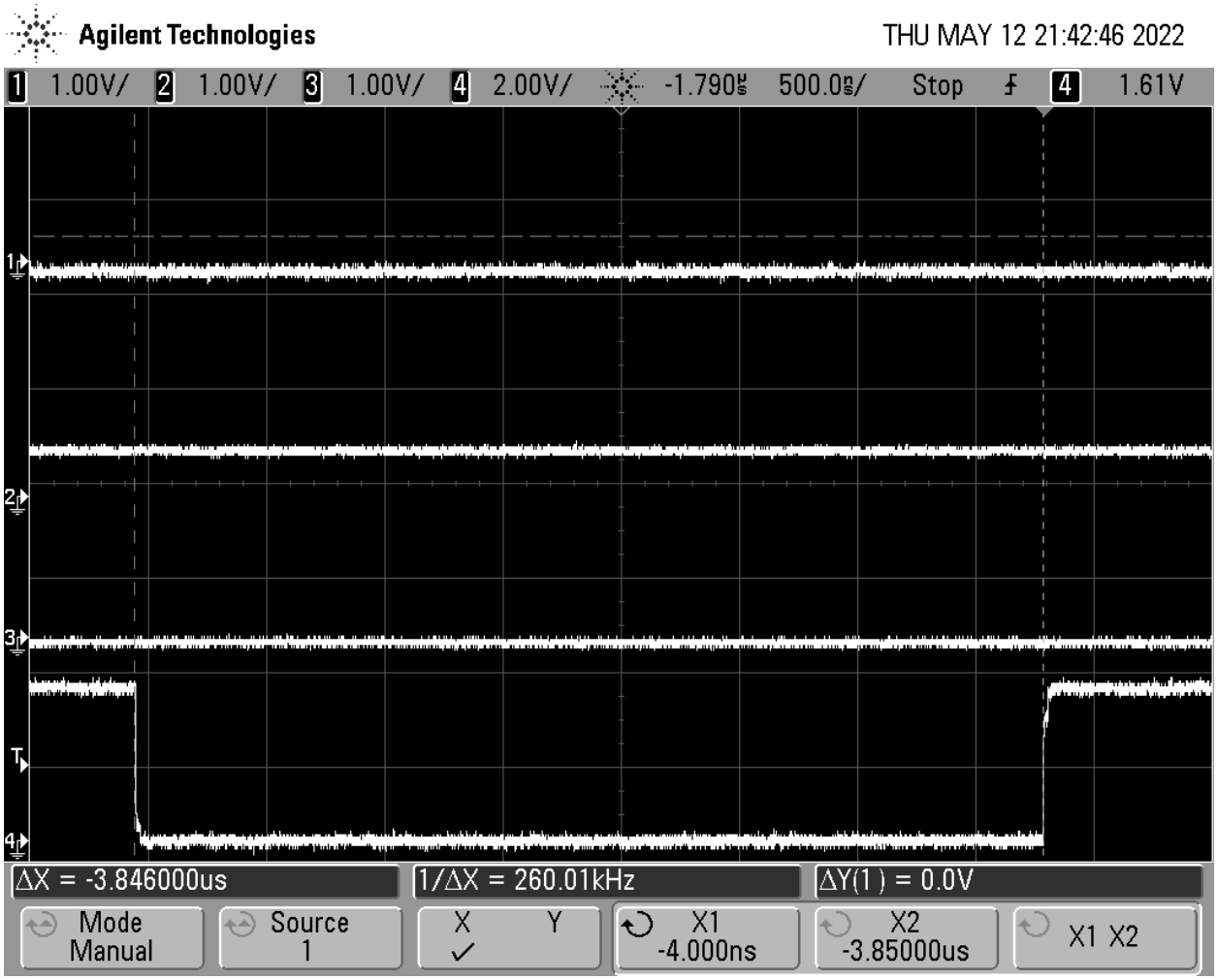


Figure 2: Line measurement with cursors marking the length (duration) of the horizontal synchronization pulse

END Subtask 1

# Task 2: Tetris Game

**Subtask 2**

Briefly describe the architecture of your tetris_game module. Are there any submodules? What is their purpose? How many FSMs did you use?

I used 2 FSMs, a small one which implements the Sound and Rumble functionallity and one for the rest of the game functions. The smaller one is bascially a Timer which waits on the change of the game_over signal or the rows_removed signal from the main FSM. The Bigger one has multiple important States; the PROCESS_INPUT State checks for player input, afterwards testing with the tetromino collider if the movement is allowed, if it is the change to the position is made. If it isn't allowed the FSM checks if the tetromino schould be placed (if the collision was while moving downward). If the tetromino is to be placed the 4 x and y coordinates get calculated with the help of the TC and saved in an array, which afterwards get written to the SRAM in the state WRITE_BLOCK_MAP_MAIN. After writing new blocks it is possible for full lines to exist, this gets checked in CHECK_IF_LINE_FULL one line at a time and then, if one is found

it is removed in DELL_LINE and then COPY_LINE_DOWN copys the blocks above the deleted line down. This process gets repeated until every line is checked. Now the tetrominos get drawn, this starts by setting the GP to x=y and y=draw_blocks_y, the second one beeing a counter for the FSM to keep track which lines it already drew. This process gets done again at the end of every line, then DRAW_BLOCKS_MAIN draws one line of blocks. The tetromino BLOCKS get saved in the block map after the info for the TC (with teromino "000" mapped to "111") The HUD is implemented with a big GFX instruction array and almost all of the code from the beginning still gets used (although most is modified). The fall down logic has its own state while most other features (like blinking or scorekeeping for example) are implemented with conditional assignments in other FSM states.

END Subtask 2

# Task 3: Bonus: SignalTap Measurement

**Subtask 3**

Trigger Condition

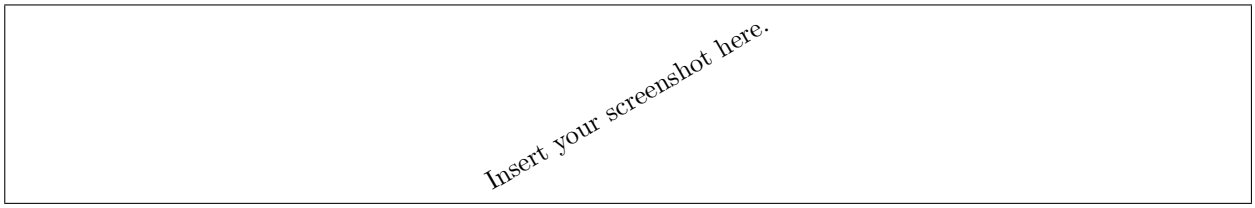


Figure 3: Screenshot showing the trigger condition

END Subtask 3

**Subtask 4**

Measurement Screenshot

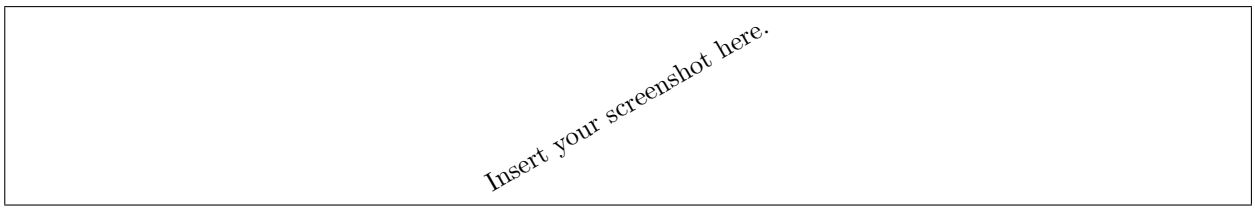


Figure 4: Screenshot showing at least the first 4 instructions (and their associated data items) issued to the graphics controller during one frame by the tetris_game module.

END Subtask 4

**Subtask 5**

Instruction Decoding

| Command | Operands | Instruction Name | Description |
|---|---|---|---|
| 0x.. | • 0x0001 | ?? | ... |
| 0x.. | • 0x0001<br>• 0x0002 | ?? | ... |
| 0x.. | • 0x0001<br>• 0x0002<br>• 0x0003 | ?? | ... |
| 0x.. | • 0x0001<br>• 0x0002<br>• 0x0003<br>• 0x0004 | ?? | ... |

END Subtask 5