CODE REVIEW

The purpose of the code review is to identify design & code problems before you spend hours testing hardware. To complete the code review:

- Project partners must identify who will own which block. (The top-level vdp block is assumed to be trivial and will not be submitted for code review).
- Project partners must identify a hard deadline for each block's material to be submitted for review, this will be used for assessment as below, so that no student can be disadvantaged by a partner not submitting code in time for the review to be carried out. It will usually be in the interests of all for this time to be kept, with if necessary less complete code submitted. This deadline depends on individual student workload. It must be agreed, and not changed, at the start of work.
- Each project partner must submit code and documentation for their block as detailed below

Documentation detail:

- FSMs must be specified by diagrams indicating states, transitions, and outputs. Significant states should be listed indicating their function. Issues not yet understood must be clearly stated. Because speed is of the essence no marks will be given for beautifully written documentation, hand drawings are acceptable for FSMs providing these are legible and scanned into electronic doucments.
- FSM inputs and outputs must be listed, with intended function.
- Processes implementing registers or combinational logic in hardware should be named, specified clocked or combinational, with list of driven outputs and description of functionality. These may in some cases be English, rather than detailed truth-table etc. Processes that make up an FSM should be listed but functionality is assumed from the FSM description, a comment in the code should reference this.

Code detail:

- Code submitted must compile, but need not synthesise or run. You must submit your initial code, not the code which is "cleaned up" after review. There is no advantage in the assessment to submitting code which has fewer errors, or which has been cleaned up by synthesis.
- Code must be complete at process & signal level for major sub-blocks, however some processes may not have complete code. Thus the functionality specified in the documentation for a process may not be implemented in code. In this case the functionality not coded should be indicated as a comment. Normally it will be possible to code everything, since there is no requirement for code to work, and writing non-working code is pretty quick.

The reviewer must check code, and report on each of the following issues. If no problems are discovered the review must state how the check was completed. Note that many problems may be revealed from synthesis messages.

Assessment of the review will depend on evidence that the reviewer understands how to check each issue, not on whether the code is good or whether mistakes are found. A null return for good code does not indicate the review has been done, so in this case the reviewer must state precisely how they checked. There is no requirement to "mend" code, however precise description of errors, where they exist, will be helpful. High quality code may result in a very boring review which identifies no errors, such a review is just as much work to write as one for low quality code because detailed evidence must be given of how each issue was checked.

Review detail:

- Synthesis: all error messages considered and either allowed or flagged for correction. Reasons must be given for decision.
- FSMs: state diagrams checked and appear to work. FSM reset state must not be first listed state in enumeration type. FSM design appears to work (any obvious problems, unreachable states, etc should be identified). FSM's can be checked by a pencil and paper "walk-through" of states during typical operation. This can also check timing of registers if the passage of data (pixel write commands etc) through registers is tracked in each state. It is not expected that every corner case can be checked manually in this way, but a check of typical operation from the documentation spec will identify obvious errors and be faster than debugging using simulation which is not expected.
- Reset: All clocked registers are reset from common signal, or else driven from signal which has known state after reset. This is especially true of FSMs.
- Code correctness: all my rules for synthesisable processes obeyed. No combinational cycles. (NB this applies also across the two sub-blocks – beware since wrong implementation of interface handshake can create cycles!). Each rule checked must be stated, even if the code presented is correct, with the reasons it is known to be ok specified, or the problems highlighted.
- Design correctness: is each signal correctly (singly) driven (clocked or combinational as stated in spec)? Remember clocked signals imply one cycle delay. Major problems in design should be highlighted.
- Code style: code which is overly long should be noted, with suggestions for improvement. Code which is badly laid out should be noted (emacs and possibly other tools will do automatic indentation on keyword capitalisation). Processes that should be merged or split should be noted (but there is some flexibility and room for individual preference here). All processes must have names.
- Comments should reference process names, and lines in listing as well if you prefer.

Review Submission

(1) One word or PDF document for each block (db or rcb), named with block name and id of student generating code and documentation for block (eg db_tjwc). The document contains the block **documentation**, **code**, and **review** in three sections as above. This submission should be prepared by the reviewer, but include copied code & documentation from reviewee.

The first sentence in the submitted document must specify which block is being reviewed, which group number, who is the reviewer, who is the reviewee. Both names and blackboard IDs should be given.

VHDL files with all code (both blocks), in compilable form, using the specified project filenames: db.vhd, rcd.vhd, project_pack.vhd. You should submit config.vhd and any other package necessary with your code. Constant, type, function, etc definitions private to one sub-block should be in the architecture declaration section of that block, not a package.

All the above files for the group submitted as a single zip (not rar) named `gp<number>` where number is your group number, e.g. `gp7.zip`, `gp22.zip`.

Assessment

It is expected that both students will cooperate over the review of both blocks. The review is group marked and no account is taken of who has actually written the review. It is expected that students will decide before work on the deliverable commences when full written material will be available for review, so that there is time for the review process to complete. If either student does not meet this agreed time their individual mark for material will be zero and the review mark wholly from the other student's material.

	Description	Max Mark
Material	Is the submitted code to be reviewed adequate in quantity, quality, and documentation (FSMs, block diagrams, etc) for the review to be conducted? No marks are provided for the correctness of the code, FSMs, quality here relates to completeness of cocumentation. Code inside processes may be omitted/wrong, however all outputs should be driven somehow to allow checking via synthesis (undriven or constant driven outputs will be optimised out together with code they drive). FSMs must be provided as state diagrams, with corresponding processes. Code for the FSMs may not be fully written. Many marks are lost through students spending too long on writing and debugging designs before the code review, for which they get no marks, and not properly documenting what they submit. Documentation before you simulate will often identify design mistakes and save time. Note also that where you realise design mistakes you need not correct these for the review deliverable. There may not be time, and you gain no marks from doing this.	7 individual (for student writing the material)
Code Review	Is there evidence in the review that all issues have been reviewed? No marks are allocated to the correctness of the design - which may have many problems. Where a block's material is submitted late it need not have a review: in this case the individual mark for the material will be zero, and the group mark for review will come from the other block. Pre-agreed deadlines may only be changed by agreement between both students in a pair.	7 group