

VOORBLAD SCHRIFTELIJKE TOETSEN

OPLEIDING	:Mechatronica
TOETSCODE	:zie toets
GROEP	:MEH2
TOETSDATUM	: 21-06-2013
TIJD	: zie toets
AANTAL PAGINA'S (incl. dit voorblad)	:voorblad + 7
DEZE TOETS BESTAAT UIT	: zie toets
GEBRUIK HULPMIDDELEN	: zie toets
TOETSOPGAVE INLEVEREN	: ja
TOEGESTANE HULPMIDDELEN	: zie toets
OVERIGE OPMERKINGEN	: Enkelzijdig afdrukken Geen papier uitdelen Kladpapier is achterzijde toetsen
OPSTELLER VAN DEZE TOETS	:Koreneef
NAAM 2^E LEZER	:Mooijekind

BELANGRIJKSTE PUNTEN UIT HET ONDERWIJS- EN EXAMENREGELING:

- je dient je via Osiris ingeschreven te hebben voor deze toets
- schrijf je naam, je studentnummer, de toetscode en de naam van de docent meteen op het tentamenpapier
- leg je identiteitsbewijs op de hoek van de tafel
- zet alle elektronische communicatiemiddelen (mobiele telefoon, PDA, etc.) uit en stop deze in je tas; deze mogen niet als calculator of klok worden gebruikt
- je mag het lokaal het eerste halfuur niet verlaten
- volg de instructies op het toetsvoorblad
- steek je hand op als je een vraag hebt

Naam:..... **Klas: MEH2** **Studentnr:**

Deze toets bestaat uit zes opgaven:

Opgaven 1 t/m 7 elk 4 punten. Opgave 8 en 9 elk 15 punten. Opgave 10 42 punten

Veel succes

Tijdens deze toets mag er geen gebruik gemaakt worden van boeken, dictaten, aantekeningen etc.

Gebruik de achterzijde van dit papier als KLADPAPIER

Voorbeeld van invullen en verbeteren MC vragen!

A	X	Fout
B		
C	X	
D		

Gecorrigeerd

Verbeterd antwoord

1.

Benoem de fase van het ontwerpproces van het controle-deel van een logische schakeling en geef de juiste procedure (stappen) aan om het te ontwerpen
 Gebruik bij de procedure cijfers, 1 moet als eerste worden uitgevoerd.

Procedure	Fase van het ontwerpproces

2. De eigenschap dat een verandering aan de ingang van een control blok, één klokpuls later resulteert in een verandering van de uitgangen, toont aan dat het control blok is opgebouwd volgens het state machine model van ...

(vul in)

Niets uit deze toets mag worden gepubliceerd of gereproduceerd anders dan bedoeld voor het afnemen van de toets zonder schriftelijke toestemming van de rechthebbende

3. Voor het weergeven van de werking van een state machine bestaan verschillende beschrijvingsmethoden. Vul onderstaande tabel in met twee beschrijvingsmethoden

1
2

4.

- I De state machine zit in het control-gedeelte van een digitale schakeling.
- II Het control-gedeelte van een digitale schakeling bestaat minimaal uit één state machine, maar kan ook meerdere state machines bevatten.

Van de twee beweringen is :

- a. I en II juist
- b. I juist en II onjuist
- c. I en II onjuist
- d. I onjuist en II juist

A		
B		
C		
D		

5. Een praktische richtlijn bij toestands codering is:
- a. minimaliseer het aantal ingangen en uitgangen
 - b. minimaliseer het aantal state-toestanden
 - c. maximaliseer het aantal state-toestanden
 - d. maximaliseer het aantal state-variabelen dat niet verandert in een groep van gerelateerde toestanden

A		
B		
C		
D		

6.

Willen we in een state machine de tijd ook een rol laten spelen in de besturing wordt aan een state machine gekoppeld :

- a. een teller die de systeemklok deelt en bepaalde tijdstippen decodeert
- b. een teller die de systeemklok vermenigvuldigt en bepaalde tijdstippen decodeert
- c. een schuifregister die naar links schuift en een bepaald bitpatroon vermenigvuldigt.
- d. Fout, aan een state machine mogen geen tellers of schuifregisters worden gekoppeld

A		
B		
C		
D		

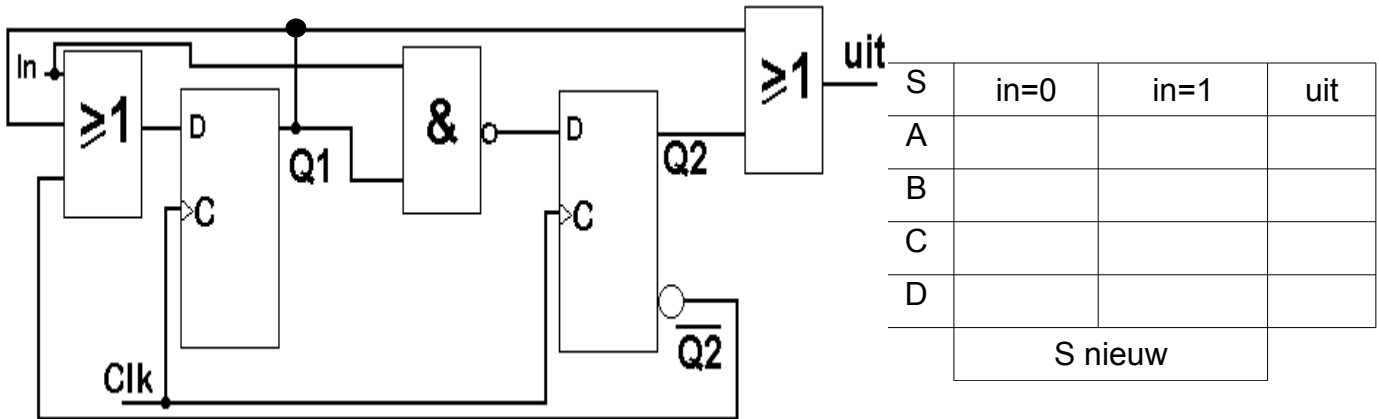
7.

Noem de drie onderdelen van een state machine:

1
2
3

Niets uit deze toets mag worden gepubliceerd of gereproduceerd anders dan bedoeld voor het afnemen van de toets zonder schriftelijke toestemming van de rechthebbende

8. Gegeven is het onderstaande schema. **Opdracht** Vul onderstaande toestandstabel in::



Q1,Q2 kunnen de waarden 00, 01,10 en 11 hebben die resp. de toestandsnamen A,B,C en D hebben. Dus B = 01 (Q1,Q2) etc. Let op: De schakeling gebruikt $\overline{Q2}$

9. Beschrijf de ingangsdecoder in VHDL van de schakeling uit opdracht 8

Uitwerking:

OPEN VRAAG 10 ZIE VOLGEND BLAD

Tijdens deze toets mag er geen gebruik gemaakt worden van boeken, dictaten, aantekeningen etc.

OPGAVE 10 (gebruik voorbedrukte tabel hieronder).

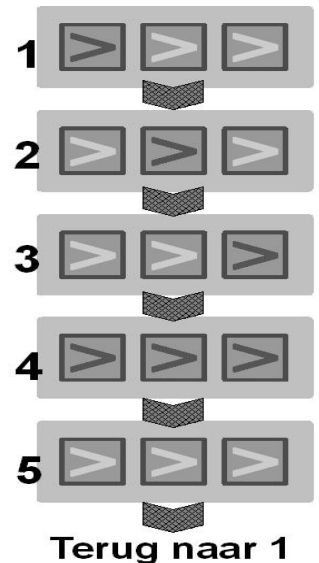


Foto: Autobus

Een autobus wil beter gezien worden in het verkeer. Om dit te bereiken wordt de autobus aan de achterzijde voorzien van twee grote richtingaanwijzers (zie foto).

Eén aan de linker- en één aan de rechterkant. Elke richtingaanwijzer heeft drie pijlen die aan- en uitgeschakeld kunnen worden.

Maak een toestandtabel van de schakeling die de richtingaanwijzers bestuurt.



Figuur

De werking

De buschauffeur geeft, met de op het stuur aanwezige schakelaar S, aan of hij rechts of links wil afslaan. Let op de schakelaar S heeft drie standen namelijk neutraal (00), rechts(01) en links(10). De bijbehorende richtingaanwijzer op de achterzijde gaat branden volgens de procedure zoals aangegeven in de figuur.

Elke toestand(1 t/m 5) duurt 1 sec. In de figuur staat rechts uitgewerkt waarbij de donkere pijl aan geeft dat het licht brandt en de licht grijze pijl aan geeft dat het licht uit is. Links werkt gelijk aan rechts alleen met de linkerverlichting.

Verder is gegeven dat het geheel werkt met een systeemklok van 1 Hz.

Opdracht 10A

Beschrijf wat je met de ongebruikte waarde 11 van de schakelaar S gaat doen en waarom ?

Geef je antwoord in onderstaand vak (alles er buiten wordt niet beoordeeld)

OPDRACHT 10B ZIE VOLGEND BLAD

Tijdens deze toets mag er geen gebruik gemaakt worden van boeken, dictaten, aantekeningen etc.


```

component ID [is
  generic ( ID : TYPEID [= expr]; )
  port ( ID : in | out | inout TYPEID [= expr]; )
end component [COMPID];

[impure] function ID
  ([constant | variable | signal] ID :
  in | out | inout TYPEID [= expr]);
return TYPEID [is
begin
  (sequential_statement)
end [function] ID];

procedure ID([constant | variable | signal] ID :
  in | out | inout TYPEID [= expr]);
[is begin
  (sequential_statement)
end [procedure] ID];

for LABELID | others | all : COMPID use
  (entity [LIBID.]ENTITYID ([ ARCHID ])|
  (configuration [LIBID.]CONFID)
  [[generic map ( (GENID => expr.) )]
  port map ( (PORTID => SIGID | expr.) )];

```

3. EXPRESSIONS

```

expression ::=
  (relation and relation) |
  (relation or relation) |
  (relation xor relation)
relation ::= sheexpr [relop sheexpr]
sheexpr ::= sheopr [shop sheopr]
sheopr ::= [+|-] term [addop term]
term ::= factor [mulop factor]
factor ::=
  (prim [* prim]) | (abs prim) | (not prim)
prim ::=
  literal | OBJID | OBJID/ATTRID | OBJID(expr.)
  | OBJID(range) | ([choice] =>) expr. |
  FCTID([PARID =>] expr.) | TYPEID'(expr) |
  TYPEID(expr) | new_TYPEID'(expr) | ( expr )
choice ::= sheopr | range | RECFCID | others

```

3.1. OPERATORS, INCREASING PRECEDENCE

```

logop      and | or | xor
relop     = | < | <= | > | >=
shop      sll | srl | sla | sra | rol | ror
addop    + | - | &
mulop    * | / | mod | rem
miscop   ** | abs | not

```

© 1985 Qualis Design Corporation. Permission to reproduce and distribute strictly verbatim copies of this document in whole is hereby granted.
 See reverse side for additional information.

```

comp_config ::=
for all | LABELID : COMPID
  (use entity [LIBID.]ENTITYID ([ ARCHID ])|
  [[generic map ( (GENID => expr.) )]
  port map ( (PORTID => SIGID | expr.) )];
[for ARCHID
  [(block_config | comp_config)];
end for. ]
end for. ]
(use configuration [LIBID.]CONFID
[[generic map ( (GENID => expr.) )]
port map ( (PORTID => SIGID | expr.) )];
end for.

```

2. DECLARATIONS

2.1. TYPE DECLARATIONS

```

type ID is ( ID. ) ;
type ID is range number downto | to number;
type ID is array ( [range | TYPEID .] )
of TYPEID | SUBTYPEID;
type ID is record
  ( ID : TYPEID; )
end record;
type ID is access TYPEID;
type ID is file of TYPEID;
subtype ID is SCALARTYPEID range range;
subtype ID is ARRAYTYPEID [range.];
subtype ID is RESOLVFCID TYPEID;
range ::=
  (integer | ENUMID | to | downto
  integer | ENUMID) | (OBJID [reverse_range] |
  TYPEID range <>)

```

2.2. OTHER DECLARATIONS

```

constant ID : TYPEID := expr;
[shared] variable ID : TYPEID [= expr];
signal ID : TYPEID [= expr];
file ID : TYPEID (is in | out string.) |
  (open read_mode | write_mode
  | append_mode is string);
alias ID : TYPEID is OBJID;
attribute ID : TYPEID;
attribute ATTRID of OBJID | others | all : class
  is expr;
class ::=
  entity | architecture | configuration |
  procedure | function | package | type |
  subtype | constant | signal | variable |
  component | label

```



0	Grouping	[]	Optional
0	Repeated		Alternative
bold	As is	CAPS	User identifier
italic	VHDL-1983		

1. LIBRARY UNITS

```

[[use clause]]
entity ID is
  generic ( ID : TYPEID [= expr]; );
  port ( ID : in | out | inout TYPEID [= expr]; );
[[declaration]]
begin
  [[parallel_statement]]
end [entity] ENTITYID;
[[use clause]]
architecture ID of ENTITYID is
  [[declaration]]
begin
  [[parallel_statement]]
end [architecture] ARCHID;
[[use clause]]
package ID is
  [[declaration]]
end [package] PACKAGE;
[[use clause]]
package body ID is
  [[declaration]]
end [package body] PACKAGE;
[[use clause]]
configuration ID of ENTITYID is
for ARCHID
  [[block_config | comp_config]]
end for;
end [configuration] CONFID;
use_clause ::=
library ID;
  [[use LIBID.PKGID.all;]]
block_config ::=
for LABELID
  [[block_config | comp_config]]
end for;

```

Tijdens deze toets mag er geen gebruik gemaakt worden van boeken, dictaten, aantekeningen etc.

SIGID'transaction[*expr*]
 Toggles if signal active
 Event on signal ?
 Activity on signal ?
 Time since last event
 Time since last active
 Value before last event
 Active driver predicate
 Value of driver
 Name of object
 Pathname of object
 Pathname to object

7. PREDEFINED TYPES

BOOLEAN True or false
 INTEGER 32 or 64 bits
 NATURAL Integers >= 0
 POSITIVE Integers > 0
 REAL Floating-point
 BIT '0', '1'
 BIT_VECTOR(NATURAL) Array of bits
 CHARACTER 7-bit ASCII
 STRING(POSITIVE) Array of characters
 TIME hr, min, sec, ms, us, ns, ps, fs
 Time => 0
 DELAY_LENGTH

8. PREDEFINED FUNCTIONS

NOW Returns current simulation time
 DEALLOCATE(ACCESS_TYP OBJ) Deallocate dynamic object
 FILE_OPEN([status], FILEID, string, mode) Open file
 FILE_CLOSE(FILEID) Close file

9. LEXICAL ELEMENTS

Identifier ::= letter { [underline] alphanumeric }
 decimal literal ::= integer [. integer] [E(+|-) integer]
 based literal ::=
 integer # hexint [. hexint] # [E(+|-) integer]
 bit string literal ::= B[IOX " ' hexint "
 comment ::= -- comment text

[LABEL:] [postponed] assert *expr*
 [report string] [severity note | warning | error | failure];
 [LABEL:] [postponed] SIGID <=
 [transpont] | [reject TIME inertial]
 { [expr [after time]] | unaffected when *expr*
 else } [expr [after time]] | unaffected;
 [LABEL:] [postponed] with *expr* select
 SIGID <= [transpont] | [reject TIME inertial]
 { [expr [after time]] |
 unaffected when choice { [choice] } };

LABEL : COMPID

[generic map ((GENID => *expr*,))]
 port map ([PORTID => SIGID,]);
 LABEL : entity [LIBID, ENTITYID] [ARCHID]
 [generic map ((GENID => *expr*,))]
 port map ([PORTID => SIGID,]);
 LABEL : configuration [LIBID,] [CONFID]
 [generic map ((GENID => *expr*,))]
 port map ([PORTID => SIGID,]);
 LABEL : if *expr* generate
 { [parallel_statement] }
 end generate [LABEL];
 LABEL : for ID in range generate
 { [parallel_statement] }
 end generate [LABEL];

6. PREDEFINED ATTRIBUTES

TYPID'base Base type
 TYPID'left Left bound value
 TYPID'right Right-bound value
 TYPID'high Upper-bound value
 TYPID'low Lower-bound value
 TYPID'pos(*expr*) Position within type
 TYPID'val(*expr*) Value at position
 TYPID'succ(*expr*) Next value in order
 TYPID'prec(*expr*) Previous value in order
 TYPID'leftof(*expr*) Value to the left in order
 TYPID'rightof(*expr*) Value to the right in order
 TYPID'ascending Ascending type predicate
 TYPID'image(*expr*) String image of value
 TYPID'value(string) Value of string image
 ARYID'left(*expr*) Left-bound of [nth] index
 ARYID'right(*expr*) Right-bound of [nth] index
 ARYID'high(*expr*) Upper-bound of [nth] index
 ARYID'low(*expr*) Lower-bound of [nth] index
 ARYID'range(*expr*) 'left downto 'right
 ARYID'reverse_range(*expr*) 'right downto 'left
 ARYID'length(*expr*) Length of [nth] dimension
 ARYID'ascending(*expr*) 'right >= 'left ?
 SIGID'delayed(*expr*) Delayed copy of signal
 SIGID'stable(*expr*) Signals event on signal
 SIGID'quiet(*expr*) Signals activity on signal

4. SEQUENTIAL STATEMENTS

wait (on (SIGID,)) [until *expr*] [for time];
 assert *expr*
 [report string] [severity note | warning | error | failure];
 report string
 [severity note | warning | error | failure];
 SIGID <= [transpont] | [reject TIME inertial]
 { [expr [after time]] };

VARID := *expr*;

PROCEDUREID([PARID => *expr*]);

[LABEL:] if *expr* then
 { [sequential_statement] }
 [elsif *expr* then
 { [sequential_statement] }]
 [else
 { [sequential_statement] }]
 end if [LABEL];
 [LABEL:] case *expr* is
 { when choice { [choice] } =>
 { [sequential_statement] } }
 end case [LABEL];
 [LABEL:] [while *expr*] loop
 { [sequential_statement] }
 end loop [LABEL];
 [LABEL:] for ID in range loop
 { [sequential_statement] }
 end loop [LABEL];
 next [LOOPLBL] [when *expr*];
 exit [LOOPLBL] [when *expr*];
 return [expression];
 null;

5. PARALLEL STATEMENTS

[LABEL:] block [is]
 [generic (ID : TYPEID,)]
 [generic map ((GENID => *expr*,))]
 [port (ID : in | out | inout TYPEID,)]
 [port map ([PORTID => SIGID | *expr*,])]
 [declaration]
 begin
 { [parallel_statement] }
 end block [LABEL];
 [LABEL:] [postponed] process [((SIGID,))]
 [declaration]
 begin
 { [sequential_statement] }
 end [postponed] process [LABEL];
 [LBL:] [postponed] PROCID([PARID =>] *expr*);

Tijdens deze toets mag er geen gebruik gemaakt worden van boeken, dictaten, aantekeningen etc.

© 1995 Qualis Design Corporation. Permission to reproduce and distribute strictly verbatim copies of this document in whole is hereby granted.
 Qualis Design Corporation
 Beaverton, OR USA
 Phone: +1-503-531-0377 FAX: +1-503-629-5525
 E-mail: info@qualis.com
 Also available: 1184 Packages Quick Reference Card
 Verilog HDL Quick Reference Card