

## Lời nói đầu năm!

Hôm nay là ngày đầu tiên của năm mới 2017 dương lịch (4/12/2016 âm lịch), nghĩa là chỉ còn 26 ngày nữa là sẽ chính thức chuyển sang năm mới dương lịch và âm lịch.

Tôi VND thăm mong và cầu chúc cho cá nhân tôi một năm mới sống và làm việc theo khẩu quyết của năm: **“SỐNG CHO ĐÁNG SỐNG”**.

Mục tiêu của năm:

- ✓ Upgrading my own CV (Curriculum Vitae)
- ✓ Starting my PhD course at SISLAB.
- ✓ Improve my own English level via IELTS test (7.0) and Online courses from Coursera.org and Futurelearn.com

**Updated by VND at 7:41 on 1<sup>st</sup> January 2017.**

**Vậy là chính thức năm mới âm lịch đã sang (updated on 5:20 28<sup>th</sup> January 2017)!**

Mình thức dậy, khởi đầu năm mới với một cơn ác mộng đó là việc Prof. Tu TX đã chuyển tên người hướng dẫn PhD cho mình sang thầy Trần Xuân Nam nào đó mà mình không hề biết. Giấc mơ tuy gọi là ác mộng nhưng nó lại vô cùng ý nghĩa, như một hồi chuông cảnh tỉnh chính bản thân mình cần luôn nỗ lực cố gắng để **đạt được mục tiêu với hiệu quả cao nhất**, và sẽ chiến thắng được bản thân **vượt qua được các yếu điểm cơ bản của chính mình** luôn gặp phải là luôn ngã khi gặp khó khăn thường hay dễ nản chí, phân tâm và thiếu năng lượng quyết tâm cao để vượt qua và chiến thắng nó. Phương pháp để chiến thắng được điểm yếu này chính là mình sẽ **sử dụng một đến hai phút hít sâu thở đều trước khi start một công việc**, công việc đã được plan sẵn tối thiểu 5 phút trước đó và **quyết tâm cao độ trước khi thực hiện**. Điều này có thể cung cấp cho mình một nguồn năng lượng đủ lớn để có thể bắt đầu công việc liên tục từ 15-30-60 phút hay thậm chí lâu hơn. Sau mỗi khoảng thời gian làm việc mình sẽ nghe nhạc hay làm việc gì đó để lấy lại cân bằng (như tập thể dục, dọn dẹp nhà cửa, hay suy nghĩ tìm các từ khóa tích cực)! Mình hoàn toàn tin mình có thể làm được và năm 2017 này sẽ là một năm thành công với cá nhân VND mình! Chúc mừng năm mới, chúc cho bản thân tôi luôn mạnh khỏe, giàu sức sống, giàu trí tuệ, giàu tính sáng tạo, tràn đầy năng lượng, nhiệt huyết để tiếp tục chiến đấu, thành công trong con đường mà tôi đã chọn! quyết tâm và chắc chắn với khẩu hiệu: **“TÔI LÀM ĐƯỢC!”**

**(updated at 3:57 29<sup>th</sup> January 2017) (updated at 9:25 am 16 December 2017).**

Bỗng dưng nhớ ra một điều vô cùng quan trọng trong năm nay mà mình cần làm được đó là **“Cải thiện hiệu năng làm việc của bản thân”**. Nâng cao hiệu suất làm việc là điều luôn cần phải được rèn luyện vì số lượng công việc và độ khó của các việc mình làm ngày càng cao, chỉ có tăng cao được hiệu suất làm việc thì mới có thể tập trung và giải quyết được vấn đề tốt, nâng cao giá trị bản thân!

**Scheduling: Practice is perfect! PRACTICE IS PERFECT!**

04:00 – 07:00: English level Improving. (IELTS 7.0 and Online courses and teaching).

08:00 – 12:00: Working for my PhD course

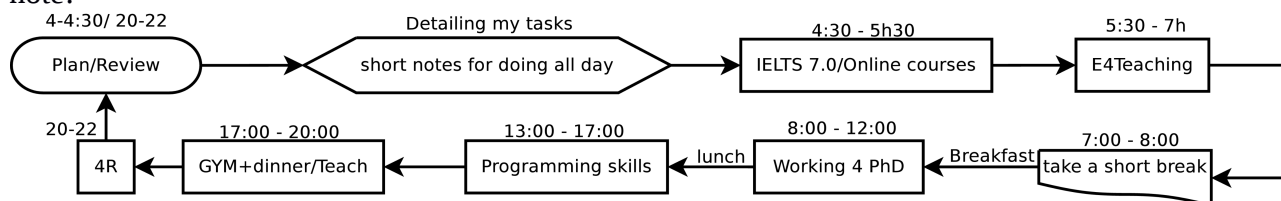
13:00 – 17:00: Programming skill (C/C++||SystemC||HDL programming).

17:00 – 20:00: GYM and dinner/ teach to 22:00.

20:00 – 22:00: 4R and do something nice and business thinking (planning for the next day).

**Updated by VND at 5:49 on 28<sup>th</sup> January 2017.**

note!



>>>>

**#Scheduling update! (on 7<sup>th</sup> July, 2017).**

04 – 05: IELTS reading (7.0 target).

05 – 05:30: Scheduling tasks!

05:30 – 7:00: Gym/ do interesting things.

7 – 8: breakfast and go to Lab.

8 – 9: discussing and warm-up section (with the college).

9 – 12: DIM<sup>1</sup> (PHD course focus)

12 – 2 pm: Read and Updating (including lunch + take a nap).

2 – 5 p.m: Skills Dev. (anything to support to my Phd course. Programming!)

5 – 9 p.m: Part-time (including dinner).

9 – 10 p.m: Review and Relax & Report.

**(Read - Review - Resume (write) - Revise - Rewrite - Report – Relax).**

**(Saturday and Sunday are spending for my wonderful family!)**

>>>>

**#story 1:** “Đơn giản hóa vấn đề bằng thái độ tích cực có lợi ích gì?” (8:46 am on Thursday 11 May 2017).

Tiêu đề của câu chuyện có vẻ hơi khó hiểu, và nó cũng khó hiểu như con người tôi vậy.

Nào cùng đơn giản hóa vấn đề với câu chuyện của tôi nhé! Let’s start!

Sáng nay, tôi nhận được một thư (email) từ khoa triển khai về việc ban hành quy chế, thủ tục... về việc thực hiện giảng dạy các môn học có nội dung là bài tập lớn!...

Tôi đã đọc và thấy: “ôi zời ơi, sao mà lại lăm thủ tục thế... bla bla bla... bạn cũng biết là thủ tục nhà nước thì nó rườm rà ntn rồi nhể!???”

Haiz, nhưng nghĩ như vậy có làm vấn đề được giải quyết ko? Tôi thiết nghĩ, mục đích của một người đi dạy như tôi là gì? Muốn đem lại lợi ích cho sinh viên, muốn tốt cho sinh viên, muốn làm cho trường ngày càng được tiếng, được miếng hơn bằng việc tạo ra những sinh viên ưu tú... blab la bla... nói tóm lại **tôi cũng thật rườm rà aaa.**

**Vậy nghĩ đơn giản là gì?**

Cùng đơn giản hóa nhé

Đứng về phía của khoa, nhà quản lý... họ muốn làm sao cho tốt nhất cho c.ty, doanh nghiệp, trường... nên việc đặt ra thủ tục... cũng chỉ là muốn tốt, và làm sao chuyên nghiệp hóa dần... OK. Nếu bạn có thể góp ý hay đàm phán về solution giúp cho quy trình hiệu quả hơn thì nhảy vô, còn không thì thông cảm và làm theo cách của bạn, just do it like me now!

Đứng về phía tôi: khi t thông suốt và nghĩ và hiểu cho hoàn cảnh của họ, t ko trách, kêu than về những thủ tục rườm rà đó, cơ bản là t làm sao để, vẫn cho SV làm BTL, cố gắng đảm bảo đánh giá khách quan, giúp SV học tốt hơn mà **không cần phải làm những thủ tục rườm rà đó!** Kaka, hay không?!! **Chuyên môn do mình nắm giữ!**

Đến đây, t chốt lại là: làm gì cũng nên đặt mình vào vị trí, hoàn cảnh của người khác để xét vấn đề được khách quan và công tâm. Còn khi có prob gì thì cứ việc nghĩ tích cực, mọi vấn đề sẽ được giải quyết tốt đẹp!

Cheer!

**Wrote by VND at 8:55 on Thursday, May 11, 2017.**

Do something which interested is much ~~more easy~~ than others!

⇒ Do something which interested yourself is much easier than others. (???)

⇒ Do something which you are interested in is much easier than others. (???)

>>>>

**#story 2: new chance, new challenge!**

Cơ hội mới, thách thức mới chính là tiêu đề cho câu chuyện của lần này!

Mình đang đứng trước một cơ hội lớn để có thể hiện thực hóa giấc mơ đi dạy thêm tiếng anh giao tiếp. Qua sự giới thiệu của Hà, một người bạn nhiệt tình năng động, mình có cơ hội tiếp cận với một số gia đình có nguyện vọng giúp các bé học Tiếng anh giao tiếp. Một tháng, một tháng để mình chuẩn bị, vẫn rất có cảm giác hoang mang lo lắng rằng không biết mình có thể đáp ứng được kỳ

---

<sup>1</sup>**DIM = Do It Myself**

vọng của các bậc phụ huynh không? Nhưng với **niềm tin mãnh liệt**, mình tin mình có thể làm được. Nhất định mình sẽ làm được.

Khó khăn trước mắt là làm sao để thi đạt được đầu vào của NCS, và chuẩn bị được một nguồn học liệu tốt để phục vụ cho tối thiểu 20 buổi học sắp tới của các bé!

Một tháng là quãng thời gian không thực sự dài, cũng không phải là ít. Mình cần tập trung để đạt được mục tiêu đề ra cho năm 2017 này! **SỐNG CHO ĐÁNG SỐNG!**

Đây có lẽ là cơ hội lớn đầu tiên trong đời mình gặp, cơ hội kiếm nhiều tiền, cơ hội tự do làm ngoài... Tất nhiên, tiền chưa bao giờ là vấn đề của mình, nhưng thực sự, mình nên nghĩ nhiều hơn về tiền, nghĩ nhiều hơn để hoàn thiện bản thân hơn, có động lực sống tốt hơn, và cơ bản là **làm việc hiệu quả hơn**.

**Updated by VND on Wednesday, May 24, 2017.**

>>>>

### **#story 3: one new day (a mess day) to me!**

Một khởi đầu khó khăn, làm sao để diễn tả được cái cảm xúc lúc này? (cảm xúc không đầu vào đâu!). Để dễ hiểu mình sẽ quay ngược lại thời gian, bắt đầu từ ngày mình đi đăng ký thi IELTS (ngày 31 tháng 5 năm 2017) để quyết đầu chuẩn đầu vào làm NCS. Lúc/thời điểm đó, mình cũng khá thoải mái và tự tin, với động lực to lớn mà thầy đưa ra “Nam đạt 6.5 thầy sẽ trả toàn bộ lệ phí thi”, uiza, nghe ngọt vậy, tận gần 5tr, số tiền 5tr tuy lớn với mình nhưng thực ra nó cũng ko là gì lắm, so với ý nghĩa mà mình có thể tưởng tượng ra. Thực sự câu nói đó của thầy làm mình khá phấn khích và quyết định cố gắng! Haizz, nhưng thực tế thì, khi bắt tay vào mới thấy khó khăn, mình làm đi làm lại khá nhiều đề thi thử, bài test trong bộ đề sách kinh điển gồm 11 cuốn của Cambridge (con số hiện tại đã cập nhật lên 12), ôi các bài làm chỉ dao động từ 4.5 – 5.5 (mà chỉ làm được hai phần nghe và đọc hiểu!). Dần dần thôi thì đã đăng ký thi cứ cố gắng thi, vì 5.0 là chuẩn đầu vào mà!

Và rồi, mọi chuyện trở nên tệ hơn và kinh khủng hơn khi vào ngày 16/6/2017, mình đăng ký thi thử tại một trung tâm anh ngữ (SCOTS) gì đó thì phải, ôi, rụng rời, kết quả thi đạt 3.5 và một lời đề nghị cho khóa học hơn một triệu được đưa ra.... Choáng váng và hoảng loạn trong một buổi sáng, và tuần sau (24/6) là bài thi của mình bắt đầu rồi! Phải làm sao với cái điểm 3.5 này đây trời? Mà mỗi khi muốn cải thiện một điểm trong band của IELTS, theo kinh nghiệm của các người đi trước cũng như chuyên gia trong ngành thì các ứng viên cần khoảng 3 tháng để cải thiện các kỹ năng... Nhưng rồi, bản thân mình tự nhủ rằng, đằng nào cũng đăng ký rồi, thôi thì có lo cũng không giải quyết được vấn đề, cố gắng tiếp tục bước đến ngày 24/6 và thi nói trong ngày 27/6 vậy.

Cái gì đến cũng đến, ngày thi đến, hôm thi 24/6 mình làm không tốt lắm, đặc biệt là phần thi đọc hiểu, và may mắn phần thi viết cảm giác như viết tốt hơn. Nghỉ một hai ngày trước khi bắt đầu buổi thi cuối là thi nói vào ngày 27/6. Ôi, hôm đó đi trễ chứ, sang chưa đến nơi thì BC gọi hỏi đang ở đâu và có thi không? Thực tế lúc đó là 9h sáng, lịch thi của mình là 9h10 và họ yêu cầu đến trước 20p. Hãi thật! Rồi vài phút khi có mặt tại BC, mình ngay lập tức được đăng ký thủ tục và đưa lên thi liền, hix, may mắn thay chủ đề không thực sự xa lạ với mình, đặc điểm khó khăn duy nhất đó chính là vẫn hơi run và sắp đặt ý chưa tốt, hơn nữa nói còn khô hết họng (do không kịp chuẩn bị nước trước khi vào thi). ...

Thi xong ai cũng hỏi thăm, mình vẫn còn ám ảnh vụ thi thử 3.5 và không dám chắc điều gì... Cảm giác lo lắng và hồi hộp chờ đợi. Rồi đến ngày 6/7, trên hệ thống bắt đầu báo đã có điểm của ngày thi 24/6 mà không tài nào tra được! Đành đợi đến sáng nay, 7/7/2017, chính thức tự phi xe đến BC hỏi thăm và lấy kết quả.

Ôi, lo lắng và run khi sợ bị trượt, nhưng thực lòng mình vẫn có niềm tin mình pass. Và cuối cùng là kết quả không uống công mình cố gắng với điểm số 5.0 trung bình và thành phần của từng kỹ năng đạt cũng sát so với mình tính lần lượt là 5.0; 4.5; 5.0 và 6.0.

Ngày hôm nay, viết lại mấy dòng lúng cùn không đầu vào đâu để chứng minh cái tâm trạng hỗn độn tại thời điểm này của mình!

Việc tiếp đến là làm sao tập trung để có thể lo cho ra được đúng deadline của khóa PHD, và gần hơn nữa là làm sao để kiếm nguồn hỗ trợ học phí sắp tới phải lo (tầm >20tr/năm), con số không phải lớn nhưng là khó khăn cho mình khi ở thời điểm hiện tại chưa kiếm được 2tr/tháng (không kể lương từ TN).

#### #Story 4: Bỗng dưng muốn ghi lại...

Nó ngồi đây, bắt đầu ghi xuống lại những gì não nó muốn nhớ... Có nhiều chuyện đã xảy ra... Người ấy xuất hiện trong cuộc đời nó, với những gì nó cảm nhận (dù chưa một lần gặp mặt...), nó cảm thấy thật ấm lòng, gần gũi... Chính bởi cái cảm xúc người đó mang lại cho nó mạnh mẽ đến vậy, nên hôm nay, nó chính thức muốn ghi lại những cảm xúc ấy!!!!

Chuyện nó muốn ghi lại hôm nay, **chính là người ấy bắt đầu đi làm thêm**, mà cố gắng đi làm thêm trong một tháng ít ỏi còn lại để đủ đk đi gặp nó... Không biết vì lý do gì, sự phụ của người ấy bỗng mất tích, và sự hoang mang, lo lắng lộ dần ra... Ngày hôm qua là một ngày không vui với nó và người ấy... với bao nhiêu áp lực (chuyện sự phụ rồi nhóm của người ấy, còn với nó, chuyện bài báo, báo cáo... giảng dạy, NCS..., gia đình), những áp lực đó đè nặng lên tâm khảm nó và người ấy... Nhưng sau cùng, khi chia sẻ rồi, lòng lại ấm hơn... và nó, và người ấy, tiếp tục lại muốn cố gắng vì nhau, vì một tương lai mà cả hai đang mong ngóng... là gặp nhau, trải nghiệm, thực tại sẽ có nhiều điều khác và khó khăn hơn trong những suy nghĩ màu hồng của nó... nhưng nó và cả người ấy nữa, cũng tự hiểu và luôn quyết không buông tay, để làm được điều không hề dễ, **đến bên nhau và cảm nhận**, như vậy... cái suy nghĩ đó... làm nó mạnh mẽ hơn, tự tin hơn... và nó tin, sẽ có ngày nó làm được, chắc chắn nó sẽ làm được, và sống thật hạnh phúc!

Updated at 9:12 AM Tuesday, October 03, 2017.

#### #Story 5:... Kết thúc một chuyện tình không tưởng!

Nói là chuyện tình thì có thể hơi hơi sao ấy?! Không biết nữa, nghĩ là vi diệu nhưng lại chẳng có gì? Chỉ là một cú lừa phỉnh tình cảm, thiếu thốn tình cảm và cần có sự quan tâm chính là tất cả những gì cho sự việc vừa rồi....

Mình đã thấy nhẹ lòng hơn khi mọi chuyện rõ ràng... Đã lâu lắm rồi mới có cảm giác yêu thương, nhớ nhưng nhiều đến vậy.

Đến nay, một trải nghiệm nữa được rút ra đó chính là cứ bình tâm, bình tĩnh... sự nóng vội có thể dẫn đến những kết quả không thể lường trước được (thường là kết quả tệ hại)!

Kỷ niệm lớn nhất có lẽ trong 5 ngày đầu tháng 11, ngày giông bão 3/11; ngày bão giông cũng 4/11 và kết thúc là ngày 5/11 ra về trong vội vã và lóng ngóng!

Updated at 5:45 AM, 22 November 2017.

#### #Story 6:... Choáng ngợp và cơ hội!!!

Với bài trình bày của Nam-Khánh Đăng... Với bao nhiêu hướng được mở ra, với bao nhiêu cơ hội được mở ra... nó tự hỏi bản thân nó liệu đây có phải là động lực, là cơ hội để nó thảo luận, trao đổi và phát triển hướng đề tài mà nó đang nghiên cứu, cố gắng hòa nhập để hoàn thiện bản thân...

Nó nghĩ là đúng, là cơ hội, nó sẽ cố gắng nắm bắt!

Updated on the morning at 9:55 am 23<sup>rd</sup> November 23, 2017.

#### #Story 7:.... Năm 2018 - cơ hội và thách thức....

Bắt đầu một năm mới với những câu từ đơn giản và ngắn gọn gồm: “Không từ bỏ, dù thấy vô cùng khó khăn như kiến thức nền (digital design), cơ bản về mã hóa... (hamming codes...)... còn chưa nắm vững,... Tuy nhiên, với tất cả sự trân thành, quyết tâm, vượt qua và thể hiện bằng **kết quả!**”

Updated on 26 February 2018 at 17:09

#### #story 8: ... Một mình... ngồi nghĩ vẫn vơ!

Ngay tại nơi đây, lúc này là 19:55, tại PTNSIS, tôi ngồi một mình, nghĩ vẫn vơ và nhìn lại cuộc sống của bản thân dưới lăng kính của chính mình.

Trong cái bối cảnh bụng hơi đói, ngồi đợi An đi làm chưa về (do buổi hội thảo đầu tiên)? Dự kiến từ trước theo tôi nhớ không nhầm là từ 6p.m đến 7h30 p.m là xong. ...

Hay do không đúng so với dự kiến kết hợp với việc mấy ae trong PTN trước khi về có chia sẻ và tâm sự về các dự án của PTN hiện tại mà làm tôi thấy có phần hoang mang, lo lắng....

Có rất nhiều tâm sự dồn nén cũng nên và khi rảnh rang (thực ra đâu có rảnh), là khi không biết làm gì thì đem chúng ra suy nghĩ và ngắm cùng xem xét lại...

Thật tù túng và khó khăn khi xung quanh mọi thứ ngày càng khó khăn hơn.... Tôi nhận ra những khó khăn chính mà bản thân đang gặp phải hiện tại có thể thứ tự bao gồm:

- Thứ nhất cần kể đến đó là việc làm nghiên cứu ở Lab, hiện tại vẫn chưa có dấu hiệu tích cực từ việc nghiên cứu, do bởi chưa có kết quả gì và đi kèm theo đó là một loạt các vấn đề như: thiếu kỹ năng làm việc, thiếu trách nhiệm với yêu cầu công việc, thiếu kiến thức nền tảng, thiếu tập trung dứt điểm công việc... Hàng loạt các vấn đề được chính bản thân nhìn nhận một cách khách quan nhất đã làm cho tôi rợn tóc gáy và đó cũng có thể là nguyên nhân sâu xa của việc ưu sầu mệt mỏi trong mấy ngày qua.
- Nguyên nhân chính thứ 2 đó chính là đạo gàn đây (từ ngày tôi đưa An về dịp giỗ bà nội và tham dự đám cưới em Cường nhà cô Hoàn), mọi người trong gia đình, đặc biệt là hai người tôi thương nhất là pama, đã tỏ ra lo lắng và phản đối quá nhiều.... Tôi tự hỏi chính bản thân mình câu hỏi là sao tôi luôn làm gia đình đau lòng, lo lắng? Sao mọi thứ tôi lại làm phức tạp hơn và làm mọi người phải khó xử? Tôi cảm thấy tự trách mình lắm! Nhưng tôi biết làm sao? Khi mà cả hai chúng tôi đều muốn tiến đến một cách tự nguyện, yêu thương chân thành và cùng nhau cố gắng (biết còn vô cùng nhiều khó khăn phía trước). Tôi không biết rằng liệu tôi quá kém cỏi và thiếu suy nghĩ không về việc này? Nhưng với tôi, việc đã lựa chọn và quyết định thì tôi luôn muốn làm được đến cuối cùng... bởi đó bản chất là thực lòng chúng tôi muốn bên nhau và cùng cố gắng, là lựa chọn của cả hai chứ không hề có ai ép buộc ai cả... Tôi không biết nên làm sao cho gia đình yên lòng ngoài việc nói dối rằng tôi không quan tâm em ấy nữa... nhưng thực tế lại khác... Thật buồn vì không dám đối mặt?! Tôi không dám đối mặt sao?? Không phải, tôi đã đối mặt và chấp nhận mọi rủi ro rồi đó chứ... chỉ là tôi không muốn mọi chuyện trở nên phức tạp và căng thẳng với gia đình, vì suy cho cùng người tôi yêu, tôi lựa chọn và quyết định đi cùng tôi cả cuộc đời vẫn là do tôi quyết...
- Vấn đề tồn đọng cuối cùng chính là vấn đề tương lai, kinh tế hiện tại cũng là vấn đề, tương lai sẽ như thế nào??? Khi tôi có thể giải quyết được hai vấn đề trên. Tôi vẫn giữ nguyên trong tâm thức câu nói *"Everything will be fine, as long as I still want to try!"*. Có lên nào tôi ơi....

Cũng đến lúc cần thêm động lực, động lực chính bản thân mình tạo ra, cùng thảo luận để cùng phát triển... Trong cuộc sống thì không có gì là dễ dàng cả... ta cần đối mặt và chiến đấu với các thách thức mà cuộc sống đem lại.... Thay vì thái độ khó chịu hay nhụt chí, hãy cố gắng hơn nữa vì giá trị bản thân ta sẽ được hình thành qua chính những kết quả mà ta đạt được sau khi đối mặt với các thách thức, khó khăn đó!

**updated by VND on 20:09 22 March 2018.**

>>>>

### **#Story no.09: My dream!**

I am wondering that should I use English or Vietnamese to write this story? Finally, I decide to use English to write it down to remember it as a point that I wanna remember!

Today (Thursday, 21<sup>st</sup> June 2018), I have to prepare for my Working Progress, Monthly meeting are going to on the tomorrow morning. My professor said this morning that "you should use mind map during your work". And that the main key for this story and I will use it to tell you about my story with the tittle "my dream".

>>>>



Nội dung:

PROGRAMMING.

# C/C++

# SystemC.

# Verilog and HDL et al.

# Matlab.

# LaTeX.

# LabVIEW and applications.

# GNUPLOT.

# PCB

# GitHubSkills

# Shellscript and Linux command.

# VIM skills

# CODEFIGHTS and GNU Make (bashScripts)

# Python

>>>

ENGLISH.

# Useful links and ref.

# Tips for IELTS.

# Phrasal verbs and nice sentences

# Common errors in Grammar and Pronunciation.

# Vocabularies, notes...

>>>

BACKGROUND AND BASIC KNOWLEDGE AND ONLINE COURSES.

# Useful links and ref.

#FPGA and VLSI

# ARM

# Computer Architecture and Organization.

# NoC et al.

# Arduino

# Digital Design (Essential basic knowledge)

# ICTU's guide to students.

# Machine Learning et al.

# Game Theory (Lý thuyết trò chơi)

# Other>>>

# Electronics and Circuits.

# Notes and general ideas

>>>

REFERENCES!

# SISLAB Infor.

# BSG PROJECT

# Books and reference.

# Useful webs.

# Useful sentences/idioms

# Abbreviation

# 3R techniques for Reference.

>>>insert reference here!

## # C/C++

>>>

### **#Pseudo code**

#link 1

<https://www.slideshare.net/DamianGordon1/pseudocode-10373156>

!#Link 2:

<https://www.wikihow.com/Write-Pseudocode>

!#Link 3:

<https://www.khanacademy.org/computing/computer-programming/programming/good-practices/p/planning-with-pseudo-code>

>>>

### **#should read to know how to think in scientist**

<http://openbookproject.net/thinkcs/python/english3e/>

>>>

#Link học C++

>>>

<https://www.learncpp.com/>

>>>

>>>

### **#Link to know how to study a Programming Language (C/C++) Faster!**

[http://www.cprogramming.com/how\\_to\\_learn\\_to\\_program.html](http://www.cprogramming.com/how_to_learn_to_program.html)

for sum.

>>>

### **# Wonderful link to study C and others.**

<http://www.learn-c.org/>

for C++ can go to:

<https://www.learn-cpp.org/>

or an other link as:

<http://www.cplusplus.com/doc/tutorial/>

We should go the above links then practice as many times as possible to master it!

>>>

### **#for coding style reference...**

<https://google.github.io/styleguide/cppguide.html#Naming>

note: should review!

>>>



**#SystemC studying.**

>>>

**#Where could we start to learn?**

!#

one of the most useful link from

<http://asic-world.com/systemc/index.html>

>>>

## # Verilog and HDL cont

>>>

### **# Link for studying HDL and more**

<http://www.asicguru.com/home/6/>

>>>

### **#Note for working on ModelSim**

!#

after sourcing file environment via `source path/env.sh`

we can use

`vsim -help | less`

!Nice!

#### **Example:**

after write an \*.do file, we can use:

`vsim -c -do *.do`

>>>

### **# Reference design for HDL**

>>>

Links:

<http://referencedesigner.com/tutorials/index.php>

>>>

### **#How to watch file .vpd**

`dve -vpd *.vpd`

>>>

## **#MODELSIM**

>>>

#one link to ref.

[http://www.tkt.cs.tut.fi/tools/public/tutorials/mentor/modelsim/getting\\_started/gsms.html](http://www.tkt.cs.tut.fi/tools/public/tutorials/mentor/modelsim/getting_started/gsms.html)

>>>

link.ref4

<https://www.mentor.com/products/fpga/verification-simulation/modelsim/>

username: email: vndprogrammings

pass: VNDprog...newnumber

>>>

## **VERILOG**

>>>

### **#Link to developing.**

[https://www.doulos.com/knowhow/verilog\\_designers\\_guide/rtl\\_verilog/](https://www.doulos.com/knowhow/verilog_designers_guide/rtl_verilog/)

>>>

### **# Difference between verilog and vhdl**

<http://www.differencebetween.net/technology/difference-between-verilog-and-vhdl/>

one more

<http://www.electronicdesign.com/what-s-difference-between/what-s-difference-between-vhdl-verilog-and-systemverilog>

>>>

### **# The best resource that I have ever seen before until now!**

Website

[https://hdlbits.01xz.net/wiki/Main\\_Page](https://hdlbits.01xz.net/wiki/Main_Page)

username: vnd\_hdl

password: normal\_newnum

>>>

## # Link reference for dw (DesignWare) synopsys

>>>

<https://www.synopsys.com/dw/buildingblock.php>

>>>

There are many examples for HDL studying including in this link!

>>>

## # Verilog Overview

>>>

This link can help the user who want to know about Verilog programming in a brief of view

[https://www.hdlworks.com/hdl\\_corner/verilog\\_ref/index.html](https://www.hdlworks.com/hdl_corner/verilog_ref/index.html)

>>>

## VHDL!

### #Some ref. For using modelsim in linux

```
1 vcom add2num.vhd
2 vcom add2numTB.vhd
3 vsim work.add2numtb
4 add wave *
5 run 200 ns
```

runsim.do file in linux

>>>

my first design with idea from Khanh N. Dang!

System block

code ref. With add2num.vhd file

Testbench with add2numTB.vhd file

Hope you enjoy it!

>>> => It should be SystemVerilog?!

>>>

### #Link ref. with vhd code

<http://esd.cs.ucr.edu/labs/tutorial/>

nice

>>>

### # a nice link to study vhd programming language (VHDL learning)

link.ref4

<https://www.nandland.com/vhdl/tutorials/index.html>

BR!

>>>

### ?# How to use a package in vhd

!#ref from:

<http://www.ics.uci.edu/~jmoorkan/vhdlref/package.html>

>>>

### ?# package file in vhd programming?

!# ref from

<https://www.nandland.com/vhdl/examples/example-package.html>.

>>>

## SYSTEMVERILOG!

>>>

### #SoC verification SystemVerilog Course

link.ref4

<https://www.udemy.com/>

>>>

---

username: van-nam

email: [vndprogrammings@gmail.com](mailto:vndprogrammings@gmail.com)

password: Dila...\_new\_number.

<https://www.udemy.com/soc-verification-systemverilog/learn/v4/overview>

it's nice!

>>>

**#nice link that could include many nice things that I need!**

>>>

<http://www.verificationguide.com/p/home.html>.

>>>

**#Udemy**

>>>

<https://www.udemy.com/>

username: vndprogrammings@

pass: Dila...newnum

>>>

**#one nice link for starting with SystemVerilog**

[https://www.doulos.com/knowhow/video\\_gallery/#anchor33](https://www.doulos.com/knowhow/video_gallery/#anchor33)

<<<

**#my doulos account**

<<<

link:

<https://www.doulos.com/knowhow/sysverilog/>

email: [vndprogrammings@gmail.com](mailto:vndprogrammings@gmail.com)

pass: vndprogrammings\_new\_num

>>>

>>>

**#One nice link to follow to study SystemVerilog**

<https://www.quora.com/How-do-I-learn-SystemVerilog-in-a-practical-way-within-three-months>

**#how to study via code online**

<https://www.edaplayground.com/>

**!#from ASIC\_WORLD**

>>>

[http://asic-world.com/systemverilog/data\\_types8.html](http://asic-world.com/systemverilog/data_types8.html)

very nice and wonderful link to start.

>>>

**#Should be check as an annual link!**

>>>

[https://www.doulos.com/knowhow/video\\_gallery/#anchor33](https://www.doulos.com/knowhow/video_gallery/#anchor33)

and one more

<https://www.doulos.com/knowhow/sysverilog/FPGA/>

>>>

>>>

**#My own story for studying SV lang.!**

First, I am reading this link <https://www.quora.com/How-do-I-learn-SystemVerilog-in-a-practical-way-within-three-months>. Then, now I am trying to look for some OOP basic knowledge to read. I also try to practice on this link <https://www.edaplayground.com/>. Now looking for the RTL sample codes to develop them. (updating on 5 January 2018).

some keynotes for ref. From reading on-line source:

link.ref4 above fig.

A few points:

1. SystemVerilog is a language that expresses concurrency and is well adapted to describe the behavior of circuits. Verilog is a subset of SystemVerilog, and really no longer exists since it is incorporated into SystemVerilog.
2. There is a misconception that knowing the syntax of an HDL (e.g., VHDL or SystemVerilog) is sufficient to do logic design and verification. Logic design and architecture is much more than that. In my book *Real Chip Design and Verification Using Verilog and VHDL* I demonstrate the various techniques in applying an HDL to do things like an asynchronous FIFO, and the design of a CPU with FSMs and with microcode.
3. The point being that there is a distinction between knowing the language (e.g., the syntax and its variations) and applying the language to perform a task.
4. Another aspect of SystemVerilog is verification. Again, SystemVerilog provides several structures to support verification, such as classes, operators, interfaces, and SystemVerilog Assertions (SVA). UVM added a layer of libraries and methodologies to bring commonalities and flexibility.
5. Thus, UVM and the application of SVA is more than just learning the syntax of the language; there is more to the story.
6. I find that learning by example is the best way; by that, I mean being exposed to lots of examples, and then digging into the details when some syntax or structure is not clear. Babies don't learn the alphabet before they learn how to speak! Just dig in, and understand the various facets and application of design and verification using SystemVerilog, rather than learning SystemVerilog syntax first and then learn how to apply the language.

Ben Cohen

<http://www.systemverilog.us/> [ben@systemverilog.us](mailto:ben@systemverilog.us)

\* [SVA Handbook 4th Edition, 2016 ISBN 978-1518681448](#)

// For 10% discount, use code 45KJT5GN @ <https://www.createspace.com/5810350>

\* A Pragmatic Approach to VMM Adoption 2006 ISBN 0-9705394-9-5

\* Using PSL/SUGAR for Formal and Dynamic Verification 2nd Edition, 2004, ISBN 0-9705394-6-0

\* Real Chip Design and Verification Using Verilog and VHDL, 2002 isbn 0-9705394-2-8

\* Component Design by Example ", 2001 ISBN 0-9705394-0-1

\* VHDL Coding Styles and Methodologies, 2nd Edition, 1999 ISBN 0-7923-8474-1

\* VHDL Answers to Frequently Asked Questions, 2nd Edition ISBN 0-7923-8115

---

<https://verificationacademy.com/forums/systemverilog/best-way-learn-systemverilog>

>>>

### #Links for ref.

<http://www.fpga4student.com/>

(should review and take a note).

>>>

### #links for study VHDL by examples

<http://esd.cs.ucr.edu/labs/tutorial/>

>>>

### #Links for studying HDL languages

<https://www.edaplayground.com/>

>>>

### #Link Pong P.Chu prof.

[http://academic.csuohio.edu/chu\\_p/rtl/sopc\\_vlog.html](http://academic.csuohio.edu/chu_p/rtl/sopc_vlog.html)

>>>

### #Should review for survey of HDL

<https://blogs.mentor.com/verificationhorizons/blog/2013/04/23/prologue-the-2012-wilson-research-group-functional-verification-study/>

>>>

### #What is the difference between behavioral and structural in vhd?

A:

These are some answers for this question

**kingslayer** ◊

Member level 4



Join Date: Jun 2010  
Location: Milano, Italy  
Posts: 78  
Helped: 25 / 25  
Points: 1,005  
Level: 7

**Re: What's the difference between behavioral and structural in vhdl?**

Dear iVenky,

when you are modeling hardware, you can write code in different ways, independent on the HDL you are using (i.e., either VHDL or Verilog). Behavioral modeling refers to a way to write code (more precisely, to model your hardware design) based on its functionality: it's like writing the algorithm that solves your problem. With structural code, on the other hand, you are connecting different parts together to get the final design.

In some way you will generally use a mixture of the twos: if you think to bottom-up approach, you first create behavioral/algorithmic code for lowest-level blocks (e.g., flip-flop) and once you move up across the abstraction levels you mix different blocks together and connect them in a structural code (e.g., shift register). This is just a simplisitic overview, but gives you an insight on the difference.

Cheers

"Fatti non foste a viver come bruti,  
ma per seguir virtute e canoscenza".

Dante Alighieri, Divina Commedia, Inferno XXVI

2 members found this post helpful.

Ref. from the link <http://www.edaboard.com/thread261191.html>

&gt;&gt;&gt;

**#Course description for VHDL programming.**<https://www.udemy.com/getting-started-with-fpga-ise-and-vhdl/>

note for dev.

&gt;&gt;&gt;

## # MATLAB

>>>

### #Keys: Matlab+HDL+LabVIEW

link1:

<https://www.mathworks.com/products/stateflow.html>

username: [vndprogrammings@gmail.com](mailto:vndprogrammings@gmail.com)

password: Dila\*\_newnum

link2:

<https://www.mathworks.com/products/hdl-coder/features.html#fpga-prototyping>

link3:

<https://www.mathworks.com/help/hdlcoder/getting-started-with-hdl-coder.html>

link4:

>>>

### #W1 contents

## Bài tập tuần 1


Cho file dữ liệu điện tim (đính kèm) dạng file .MAT chứa dữ liệu điện tim và tần số lấy mẫu fs

Hãy

1. Vẽ tín hiệu trên miền thời gian
2. Vẽ phổ biên độ tín hiệu này

 ecg.mat

## Submission status

Submission status	Submitted for grading
Grading status	Not graded
Due date	Wednesday, 7 March 2018, 12:00 AM
Time remaining	Assignment was submitted 1 day 8 hours early
Last modified	Monday, 5 March 2018, 3:33 PM
File submissions	 EXW1-results.zip

solution!

```
clear; clc; close all;%clear all variables and windows
```

```
disp('Exercise for week 1!');
```

```
disp('Student name: Van-Nam Dinh');
```

```
FileName = 'ecg.mat';%filename that wanna load
```

```
%FolderName = '/home/userdata/hvch/namdv/MatlabCode/EXW1-results';%datapath
```

```
FolderName = '/home/userdata/hvch/namdv/WORK/source-codes/MatlabCode';%datapath
```

```
File = fullfile(FolderName, FileName);%add datapath&file name
```

```
load(File); % load file ecg.mat
```

```
%after doing this we have ecgOriginal and fs values
```

```
%1-plot the data in time domain of ecg signal
```



```

x=ecgOriginal;
sizeofx=size(x,2);
Ts=1/fs;
t = 0: Ts: (Ts*(2500-1));
%t=Ts*(0:2499);
plot(t,x); xlabel('time - s'); ylabel('ecg magnitude'); title('tin hieu ecg trong mien thoi gian');
%2-Pho bien do cua tin hieu trong mien tan so
X = fft(x,sizeofx); %Pho tan so
Xabs = abs(X(1:sizeofx/2));%Pho bien do
figure;
f=linspace(0,fs/2,sizeofx/2);
plot(f,Xabs);xlabel('freq. - Hz'); ylabel('ecg magnitude'); title('tin hieu ecg trong mien tan so');
>>>
>>>

```

## # subplot in Matlab

#! ref. link.

<http://matlab.izmiran.ru/help/techdoc/ref/subplot.html>

>>>

Week 2:

## Bài tập tuần

Dựa vào tín hiệu ECG sạch ở tuần trước, hãy

- 1) Cộng thêm tín hiệu có tần số 0,5 Hz và vẽ
- 2) Cộng thêm tín hiệu có tần số 50 Hz và vẽ
- 3) Cộng thêm tín hiệu có tần số 0,5 Hz và 50 Hz và phân tích phổ
- 4) Dùng FDAtool thiết kế bộ lọc loại bỏ thành phần 50Hz và 0,5Hz trong tín hiệu bị nhiễu

## Submission status

Submission status	No attempt
Grading status	Not graded
Due date	Thursday, 22 March 2018, 12:00 AM
Time remaining	2 days 14 hours
Last modified	Monday, 19 March 2018, 9:23 AM
Submission comments	► <a href="#">Comments (0)</a>

Add submission

Make changes to your submission

>>>

submission

>>>

Week 3:

Requirements:

## Bài tập tuần

Thực hiện ghi âm giọng nói của mình sử dụng Matlab (tập lệnh wavrecord, wavread, wavwrite,...) lưu file dưới dạng wav và MAT.

Vẽ tín hiệu trên miền thời gian, tần số

Khuếch đại tín hiệu có tần số từ 1 Khz trở lên để thu được giọng cao hơn

Lưu file chỉnh sửa dưới dạng wav và MAT.

## Submission status

Submission status	No attempt
Grading status	Not graded
Due date	Wednesday, 28 March 2018, 12:00 AM
Time remaining	14 hours 45 mins
Last modified	Thursday, 22 March 2018, 8:28 AM
Submission comments	<a href="#">Comments (0)</a>

Add submission

Make changes to your submission

submit!

>>>

>>>

### # How to read .wav file in matlab

!#

keys: wavread

link4ref.

<https://www.mathworks.com/help/matlab/ref/audioread.html>

>>>

### # How to write an audio file in matlab (wavwrite function)

!# reference from:

<http://matlab.izmiran.ru/help/techdoc/ref/wavwrite.html>

other link:

<https://www.mathworks.com/help/matlab/ref/audiowrite.html>

!nice!

>>>

### # function wavrecord

!# ref. from.

<http://radio.feld.cvut.cz/matlab/techdoc/ref/wavrecord.html>

>>>

The best way to learn something is practice. If you spend your time to practice that mean you are accumulating your value to jump to the your next level.

>>>

## # LaTeX content

>>>

### # Xử lý hiện tượng margine của itemize (tránh thụt vào sâu)

keys: [leftmargin=\*]; \usepackage{enumitem}...

link:

<https://tex.stackexchange.com/questions/91124/itemize-removing-natural-indent>

>>>

### #IEEE templates for users!

<http://ieeauthorcenter.ieee.org/create-your-ieee-article/use-authoring-tools-and-ieee-article-templates/ieee-article-templates/>

>>>

### #Link TV

>>>

<http://math2it.com/tag/tu-hoc-latex/>

Mr.Tho suggested!

>>>

### #Springer's templates

<https://www.springer.com/gp/livingreviews/latex-templates>

>>>

### #Table in LaTeX

!#

Link 1 để tham khảo

<https://www.sharelatex.com/learn/Tables>

Link 2 để tham khảo

<https://en.wikibooks.org/wiki/LaTeX/Tables>

Một số mẫu codes tham khảo

```
\begin{tabular}{l c r }
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
\end{tabular}
```

1	2	3
4	5	6
7	8	9

t2

```
\begin{tabular}{l | c | r }
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
\end{tabular}
```

1	2	3
4	5	6
7	8	9

t3

```
\begin{tabular}{l | c | r }
\hline
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
\hline
\end{tabular}
```

1	2	3
4	5	6
7	8	9

t4

t5

```

\begin{center}
\begin{tabular}{l | c | r }
\hline
1 & 2 & 3 \\ \hline
4 & 5 & 6 \\ \hline
7 & 8 & 9 \\ \hline
\end{tabular}
\end{center}

```

1	2	3
4	5	6
7	8	9

t6

```

\begin{tabular}{r|l}
\hline
7C0 & hexadecimal \\
3700 & octal \\ \cline{2-2}
11111000000 & binary \\ \hline \hline
1984 & decimal \\ \hline
\end{tabular}

```

7C0	hexadecimal
3700	octal
11111000000	binary
1984	decimal

>>>

**# Tự động xuống dòng trong bảng (multiple line in latex)**

#!

```

\begin{table} [h]
{\renewcommand\arraystretch{1.25}
\begin{tabular}{|l|l|l|} \hline
Year:& \multicolumn{2}{l|}{Sự kiện} \\ \hline \hline
Scenarie:& \multicolumn{2}{p{4cm}|}{\raggedright At
oprette en server med bestemte regler som tillader folk at spille
sammen. More Text more text More Text} \\ \hline
\end{tabular}}
\end{table}

```

<<<

**#How to insert an image into LaTeX**

#!

Link for ref.

[https://www.sharelatex.com/learn/Inserting\\_Images](https://www.sharelatex.com/learn/Inserting_Images).

My own implement

```

\usepackage{graphicx}
\graphicspath{ {images/} }
\begin{document}

```

The universe is immense and it seems to be homogeneous,  
in a large scale, everywhere we look at.

```

\includegraphics{universe}

```

There's a picture of a galaxy above  

```

\end{document}

```

>>>

**Note:** the “universe” is the name of the file that you want to insert into the latex file.  
The {images/} is the folder’s name that include all files wanna to use in the TEX file.  
In practice, sometime your file can be too large, it can be fixed the size by using the following:

```
\includegraphics[scale=0.4]{universe}
```

You can also change the optimal size like that

```
\includegraphics[width=10 cm, height=4cm]{wp}
```

>>One other way to use following code

```
\begin{figure}[h]
  \centering
  \includegraphics[width=1\textwidth]{wp}
  \caption{time table for doing thesis' tasks}
  \label{fig1:}
\end{figure}
```

BR!

>>>

### # Kỹ năng chèn ảnh esp, pdf (ảnh vector) vào trong latex?

>>>

Một vấn đề với LaTeX là khi làm việc với bản thì rất khó để có thể tạo được một bảng như ý muốn trong một thời gian cho phép. Do vậy, ở đây trình bày cách tạo bảng nhanh chóng có sẵn tool trong hệ điều hành ubuntu.

**Bước 1:** Bạn tạo bảng với tool LibreOffice Calc (Tool này dùng tương tự như trong excel của Wins) nên bạn hãy tự mò và tạo cho chính bạn các bảng coi chuyên nghiệp nhất có thể nhé!

**Bước 2:** Sau khi tạo xong bảng bạn cần rồi thì việc xuất ra file ảnh định dạng vector bằng cách: chọn phần muốn xuất (chọn toàn bộ bảng). Sau đó nhấn ctrl+c để thực hiện thao tác copy dữ liệu; tiếp theo, vào file >> new >> drawing; Thao tác tiếp theo là nhấn ctrl+shift+v (nhớ chọn GDI metafile) nhé!

**Bước 3:** Xuất (export) sang định dạng vector mong muốn (eps hay pdf) tùy bạn!

Chúc may mắn!

>>>

### #Cite figure trong LaTeX?

!#

Sau khi đã tạo được (chèn được hình vào trong LaTeX, việc cite đến hình đó sẽ được thực hiện bằng cách dùng lệnh ~\ref{} là sẽ tự động chèn được những phần hình mà muốn cite.

Tham khảo từ nguồn:

<https://alvinalexander.com/blog/post/latex/reference-figure-or-table-within-latex-document>

### #Cite equation in LaTeX?

!#

You can follow the figure:

keys: \label{}; \ref{}

almost of the above commands are automatically when we use LaTeX to gen.

```
\section{Math references} \label{mathrefs}
As mentioned in section \ref{introduction}, different elements can
be referenced within a document
```

```
\subsection{powers series} \label{subsection}
```

```
\begin{equation} \label{eq:1}
\sum_{i=0}^{\infty} a_i x^i
\end{equation}
```

The equation \ref{eq:1} is a typical power series.

## 2.1 powers series

$$\sum_{i=0}^{\infty} a_i x^i \quad (1)$$

The equation 1 is a typical power series.

nice!

>>>

## #multiple references and cites in LaTeX

!#

You can ref. As

### Multiple references in LaTeX



I'm writing a paper and using the `IEEEtran.cls` in LaTeX. I need to have my references to appear like:

33

```
\cite{test1,test2,test3}
I want : [1-3]
```



I searched and found that I need to include cite package. However, after I have used cite packages... I got :



6

```
[1]-[3]
```

How can I get [1-3] instead of [1]-[3]? without using new commands ?

Here is an example of what I have :

```
\documentclass[conference]{IEEEtran}
\usepackage{cite}
\begin{document}
\section{Test}
\cite{test1,test2,test3}.
\bibliographystyle{IEEEtran}
\bibliography{test}
\end{document}
```

the test.bib file:

```
@MISC{test1,
title = {Reference test1}
}
@MISC{test2,
title = {Reference test2}
}
@MISC{test3,
title = {Reference test3}
}
```

output :

```
I. TEST
[1]-[3].
REFERENCES
[1] "Reference test1."
[2] "Reference test2."
[3] "Reference test3."
```

```
{citing} {ieeetran}
```

link ref from: <https://tex.stackexchange.com/questions/98595/multiple-references-in-latex>

one more link for testing

[https://www.sharelatex.com/learn/Bibliography\\_management\\_in\\_LaTeX](https://www.sharelatex.com/learn/Bibliography_management_in_LaTeX)

note! :( Nhìn link trên này chưa test mà thấy unfamiliar :)

>>>

## #Một vài công thức toán học như tích phân, giới hạn trong LaTeX?

!#

Tham khảo link sau:

[https://www.sharelatex.com/learn/Integrals\\_sums\\_and\\_limits](https://www.sharelatex.com/learn/Integrals_sums_and_limits)

hình ảnh mô tả

note: chú ý với gõ công thức thông thường thì để trong cặp dấu \$

ví dụ  $\alpha$  =>  $\alpha$ .

### Limits

Limit expression can be added using the

$\lim_{\text{lower}}$

command.

LaTeX code	Output
Limit $\lim_{x\rightarrow\infty} f(x)$ inside text	Limit $\lim_{x\rightarrow\infty} f(x)$ inside text
$\lim_{x\rightarrow\infty} f(x)$	$\lim_{x\rightarrow\infty} f(x)$

[Open an example in ShareLaTeX](#)

### Integer and sum limits improvement

In *inline* math mode the integral/sum/product lower and upper limits are placed right of integral symbol. Similar is for limit expressions. If you want the limits of an integral/sum/product to be specified above and below the symbol in *inline* math mode, use the  $\limits$  command before limits specification.

LaTeX code	Output
Integral $\int_a^b x^2 dx$ inside text	Integral $\int_a^b x^2 dx$ inside text
Improved integral $\int\limits_a^b x^2 dx$ inside text	Improved integral $\int_a^b x^2 dx$ inside text
Sum $\sum_{n=1}^{\infty} 2^{-n} = 1$ inside text	Sum $\sum_{n=1}^{\infty} 2^{-n} = 1$ inside text
Improved sum $\sum\limits_{n=1}^{\infty} 2^{-n} = 1$ inside text	Improved sum $\sum_{n=1}^{\infty} 2^{-n} = 1$ inside text

Moreover, adding  $\displaystyle$  beforehand will make the symbol large and easier to read.

On the other hand,  $\mathlarger$  command (provided by  $\text{relsize}$  package) is used to get bigger integral symbol in display.

```
 $\int\frac{1}{2}dx - \mathlarger{\int\frac{1}{2}dx}$ 
```

$$\int \frac{1}{2} dx - \int \frac{1}{2} dx$$

it work!

>>>

## #Chuyển tên mặc định “figure” => “Hình”

chỉ cần thêm lệnh như sau:

```
\renewcommand{\figurename}{Hình}
```

tham khảo từ nguồn:

<https://tex.stackexchange.com/questions/17489/change-caption-name-of-figures>

>>>

## #Highlight in LaTeX



#!

**ex1:**

```
\textcolor{red} {\hl{\textbf{Still reading comprehensive and  
making a decision to do for the next stage!}}}
```

**ex2:**

```
\textcolor{blue}{\textbf{\hl{All of the above topics will be  
conclude and take a short summary after understanding}}.}
```

Link ref:

<https://tex.stackexchange.com/questions/141569/highlight-textcolor-and-boldface-simultaneously/319000>

other link:

<https://www.sharelatex.com/learn/Bold, italics and underlining>

>>>

**#LaTEX Math symbol**

link.ref4

<http://web.ift.uib.no/Teori/KURS/WRK/TeX/symALL.html>

other link

<https://www.sharelatex.com/learn/List of Greek letters and math symbols>

>>>

?#

!#

>>>

#!

**#How to insert a referent link in LaTeX?**

#!

keys: hyperlinks – shareLaTeX; LaTeX Hyperlinks.

command!

```
\usepackage{hyperref} => \href{link URL} {TEXT Dis}; \url{}
```

link ref:

<https://www.sharelatex.com/learn/Hyperlinks>

chú ý: với trường hợp muốn cite ở reference thì có thể tham khảo thêm link sau:

<https://tex.stackexchange.com/questions/3587/how-can-i-use-bibtex-to-cite-a-web-page>

Một số đoạn code mẫu

Minh họa qua hình

A simple way of doing it in BibTeX is with a `@misc` entry:

```
@misc{WinNT,
  title = {{MS Windows NT} Kernel Description},
  howpublished = {\url{http://web.archive.org/web/20080207010024/http://www.808mult
  note = {Accessed: 2010-09-30}}
}
```

You should also perhaps include an author if you know it. And remember to load a package such as `hyperref` or `url`.

If you are using BibLaTeX there is an `@online` entry type:

```
@online{WinNT,
  author = {MultiMedia LLC},
  title = {{MS Windows NT} Kernel Description},
  year = 1999,
  url = {http://web.archive.org/web/20080207010024/http://www.808multimedia.com/win
  urldate = {2010-09-30}}
}
```

đoạn code cụ thể:

```
@misc{WinNT,
  title = {{MS Windows NT} Kernel Description},
  howpublished =
{\url{http://web.archive.org/web/20080207010024/http://www.808mult
imedia.com/winnt/kernel.htm}},
  note = {Accessed: 2010-09-30}}
}
```

mẫu 2:

```
@online{WinNT,
  author = {MultiMedia LLC},
  title = {{MS Windows NT} Kernel Description},
  year = 1999,
  url =
{http://web.archive.org/web/20080207010024/http://www.808multimedi
a.com/winnt/kernel.htm},
  urldate = {2010-09-30}}
}
```

với mẫu 2 thì là sử dụng BibLatex online

>>>

>>>

## #Cách đánh dấu link trong LaTeX.

!# Một mẹo nhỏ là dùng label bằng cách sau:

- ✓ B1: tại nơi bạn muốn đánh dấu để link lại đó dùng lệnh `\label{key}`; Trong đó *key* sẽ là khóa để bạn link lại trong bước 2.

- ✓ B2: Tại nơi bạn muốn link quay lại chỗ cần ref bạn chỉ cần dùng lệnh `\ref{key_list}`; một `key_list` hiện ra và bạn chỉ việc chọn đúng `key` mà bạn cần link là sẽ ổn.

BR!

Một mẹo nhỏ nữa để chèn (insert) nhanh địa chỉ link web bạn dùng lệnh `\url{URL}`; trong đó URL là link bạn muốn chèn vào.

>>>

### # Đoạn code hỗ trợ viết tiếng việt trong LaTeX

!#

```
\documentclass{report}
\usepackage[T5]{fontenc}
\usepackage[utf8]{inputenc}
\begin{document}
Tiếng Việt
\end{document}
```

Tham khảo từ nguồn:

<https://nngocchan.wordpress.com/2011/03/14/vietnamese-in-latex/>

!#

>>>

### #How to change the font in LaTeX?

!#

You can use the following link:

[https://www.sharelatex.com/learn/Font\\_typefaces](https://www.sharelatex.com/learn/Font_typefaces)

a simple code

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage{geometry}
\geometry{textwidth=7cm}

\usepackage{tgbonum}

\begin{document}
This document is a sample document to
test font families and font typefaces.

{\fontfamily{qcr}\selectfont
This text uses a different font typeface
}
\end{document}
```

This document is a sample document  
to test font families and font typefaces.  
This text uses a different font  
typeface

!nice!

>>>

### #Một link tham khảo về tài nguyên loại font trong latex.

<http://www.tug.dk/FontCatalogue/>

! Cần đọc thêm và thực hiện thử vì chưa có time kiểm chứng...

>>>

### #Highlight in LaTeX

!#

can use the following ref. Codes

```
\documentclass{report}
\usepackage{color,soul}
\begin{document}
  \hl{foo}
  \hl{\textbf{foo}}
  \textcolor{red}{foo}
  \textbf{\textcolor{red}{\hl{foo}}}
  \textcolor{red}{\textbf{\hl{foo}}}
\end{document}
```

Link ref.

<https://tex.stackexchange.com/questions/141569/highlight-textcolor-and-boldface-simultaneously>

>>>

>>>

## # LabVIEW and Applications

>>>

### **#How to develop LabVIEW environment?**

!#

keys: LabVIEW github; LabVIEW projects...

Link ref:

### **#Links for Dev. Projects.**

Link1: electronicsforu: should be followed frequently.

<http://electronicsforu.com/?s=labview>

>>>

vnd1711117

## # GNUPLOT and notes

>>>

### **#How to export output in gnuplot**

>>>

#! Link ref.

<http://xmodulo.com/how-to-export-gnuplot-output-to-png-jpg-and-pdf.html>

!Notes:

it is very useful and run with png.

### **#How to run script of gnuplot**

#!

>>>

- use code and save it with \*.gnu
- use command: `gnuplot *.gnu`

>>>

### **#Plot functions from datafile ref.**

#!

link ref.

<http://www.gnuplotting.org/tag/functions/>

>>>

>>>

## **# PCB**

**#keys:** eagle, altium, ...

**#link for studying eagle (tutorial videos)**

<https://cadsoft.io/tour/>

>>>

**#Synopsys Reference Designer Tutorial...**

>>>

link nice

<https://www.synopsys.com/>

>>>

link news

<https://www.synopsys.com/implementation-and-signoff.html>

>>>

One nice link!

<http://referencedesigner.com/tutorials/index.php>

>>>

**#For integrity signal design**

>>>

[http://referencedesigner.com/tutorials/si/si\\_01.php](http://referencedesigner.com/tutorials/si/si_01.php)

it should be read carefully.

>>>

>>>

>>>



## #GitHub skills

>>>

**#big projects and link for developing**

BSG prject

**#link 1: get the source-codes of bsg\_manycore**

<https://bitbucket.org/account/user/taylor-bsg/projects/PROJ>

>>>

**#link 2: related links of bsg\_manycores doc.**

>>>

<http://bjump.org/manycore/>

>>>

**#simple skills in git:**

<https://gist.github.com/hofmannsven/6814451>

>>>

**#one way to up2date all files/folders in your master github**

```
git add .
git commit -m "August182018"
git pull origin
git push origin master
```

>>>

Note: myown github account: van-namdinh

password: normal!

>>>

**#at the beginning**

look at the following pic.

```
1617 git add install_vim.sh
1618 git commit -m "Fix bug install_vim."L
1619 git commit -m "Fix bug install_vim."
1620 git config --global user.email "vndprogrammings@gmail.com"
1621 git config --global user.name "van-nam dinh"
1622 git commit -m "Fix bug install_vim."
1623 git push origin master
```

>>>

>>>**#one of the link for ref.**

You can clone this one to read more for necessary commands.

<https://github.com/bpassos/git-commands>

one more!

<https://confluence.atlassian.com/bitbucketserver/basic-git-commands-776639767.html>

one more!

>>>

**#very useful note for uploading in GitHub**

When you wanna download something you have just use the terminal to clone it via:

`git clone git_hub_data_path`

then you should see .git in that directory after cloning it. (use command ls -la to see all files.

Use the follow commands for submit your data on that repository

```
1565 git add install_vim.sh
1566 vim install_vim.sh
1567 c
1568 git add install_vim.sh
1569 git commit -m "Fix bug install_vim."L
1570 git commit -m "Fix bug install_vim."
1571 git config --global user.email "vndprogrammings@gmail.com"
1572 git config --global user.name "van-nam dinh"
1573 git commit -m "Fix bug install_vim."
1574 git push origin master
```

## #Github and something nice with git help?

```
vnd@vnd:~$ git help
usage: git [--version] [--help] [-C <path>] [-c name=value]
         [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
         [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
         [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
         <command> [<args>]
```

These are common Git commands used in various situations:

start a working area (see also: `git help tutorial`)

<code>clone</code>	Clone a repository into a new directory
<code>init</code>	Create an empty Git repository or reinitialize an existing one

work on the current change (see also: `git help everyday`)

<code>add</code>	Add file contents to the index
<code>mv</code>	Move or rename a file, a directory, or a symlink
<code>reset</code>	Reset current HEAD to the specified state
<code>rm</code>	Remove files from the working tree and from the index

examine the history and state (see also: `git help revisions`)

<code>bisect</code>	Use binary search to find the commit that introduced a bug
<code>grep</code>	Print lines matching a pattern
<code>log</code>	Show commit logs
<code>show</code>	Show various types of objects
<code>status</code>	Show the working tree status

grow, mark and tweak your common history

<code>branch</code>	List, create, or delete branches
<code>checkout</code>	Switch branches or restore working tree files
<code>commit</code>	Record changes to the repository
<code>diff</code>	Show changes between commits, commit and working tree, etc
<code>merge</code>	Join two or more development histories together
<code>rebase</code>	Forward-port local commits to the updated upstream head
<code>tag</code>	Create, list, delete or verify a tag object signed with GPG

collaborate (see also: `git help workflows`)

<code>fetch</code>	Download objects and refs from another repository
<code>pull</code>	Fetch from and integrate with another repository or a local branch
<code>push</code>	Update remote refs along with associated objects

'`git help -a`' and '`git help -g`' list available subcommands and some concept guides. See '`git help <command>`' or '`git help <concept>`' to read about a specific subcommand or concept.

#nice!

>>>

#one picture that shows all keys for developing git!

# with the `git help -a` command.

```
vnd@vnd:~$ git help -a
usage: git [--version] [--help] [-C <path>] [-c name=value]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        <command> [<args>]

available git commands in '/usr/lib/git-core'

  add
  add--interactive
  am
  annotate
  apply
  archive
  bisect
  bisect--helper
  blame
  branch
  bundle
  cat-file
  check-attr
  check-ignore
  check-mailmap
  check-ref-format
  checkout
  checkout-index
  cherry
  cherry-pick
  clean
  clone
  column
  commit
  commit-tree
  config
  count-objects
  credential
  credential-cache
  credential-cache--daemon
  credential-store
  daemon
  describe
  diff
  diff-files
  diff-index
  diff-tree
  difftool
  difftool--helper
  fast-export
  fast-import
  fetch
  fetch-pack
  filter-branch
  fmt-merge-msg
  for-each-ref
  format-patch
  fsck
  fsck-objects
  gc
  get-tar-commit-id
  grep
  hash-object
  help
  http-backend
  http-fetch
  http-push
  imap-send
  index-pack
  init
  init-db
  instaweb
  interpret-trailers
  log
  ls-files
  ls-remote
  ls-tree
  mailinfo
  mailsplit
  merge
  merge-base
  merge-file
  merge-index
  merge-octopus
  merge-one-file
  merge-ours
  merge-recursive
  merge-resolve
  merge-subtree
  merge-tree
  mergetool
  mktag
  mktree
  mv
  name-rev
  notes
  pack-objects
  pack-redundant
  pack-refs
  patch-id
  prune
  prune-packed
  pull
  push
  quiltimport
  read-tree
  rebase
  receive-pack
  reflog
  relink
  remote
  remote-ext
  remote-fd
  remote-ftp
  remote-ftp
  remote-https
  remote-testsvn
  repack
  replace
  request-pull
  rerere
  reset
  rev-list
  rev-parse
  revert
  rm
  send-pack
  sh-i18n--envsubst
  shell
  shortlog
  show
  show-branch
  show-index
  show-ref
  stage
  stash
  status
  stripspace
  submodule
  submodule--helper
  subtree
  svn
  symbolic-ref
  tag
  unpack-file
  unpack-objects
  update-index
  update-ref
  update-server-info
  upload-archive
  upload-pack
  var
  verify-commit
  verify-pack
  verify-tag
  web--browse
  whatchanged
  worktree
  write-tree

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
```

>>>

# with the git help command:  
nice!

```

usage: git [--version] [--help] [-C <path>] [-c name=value]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
    clone      Clone a repository into a new directory
    init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
    add        Add file contents to the index
    mv         Move or rename a file, a directory, or a symlink
    reset      Reset current HEAD to the specified state
    rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
    bisect     Use binary search to find the commit that introduced a bug
    grep       Print lines matching a pattern
    log        Show commit logs
    show       Show various types of objects
    status     Show the working tree status

grow, mark and tweak your common history
    branch     List, create, or delete branches
    checkout   Switch branches or restore working tree files
    commit     Record changes to the repository
    diff       Show changes between commits, commit and working tree, etc
    merge      Join two or more development histories together
    rebase     Reapply commits on top of another base tip
    tag        Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
    fetch      Download objects and refs from another repository
    pull       Fetch from and integrate with another repository or a local branch
    push       Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.

```

For example you can type

`git help commit`

then just look at screen and read carefully to follow the guide!

BR!

>>>

**#How to upload with git hub?**

#!

\$ git add .

\$ git commit -m "Add existing file"

\$ git push origin your-branch

You should read in detail via the following pic. ?

- 1 On your computer, move the file you'd like to upload to GitHub into the local directory that was created when you cloned the repository.
- 2 Open Terminal.
- 3 Change the current working directory to your local repository.
- 4 Stage the file for commit to your local repository.

```
$ git add .  
# Adds the file to your local repository and stages it for commit. To  
unstage a file, use 'git reset HEAD YOUR-FILE'.
```

- 5 Commit the file that you've staged in your local repository.

```
$ git commit -m "Add existing file"  
# Commits the tracked changes and prepares them to be pushed to a remote  
repository. To remove this commit and modify the file, use 'git reset --soft  
HEAD~1' and commit and add the file again.
```

- 6 Push the changes in your local repository to GitHub.

```
$ git push origin your-branch  
# Pushes the changes in your local repository up to the remote repository  
you specified as the origin
```

>>>

**#Quick setup for git (should be practice)!**

!#

### Quick setup — if you've done this kind of thing before

or   `https://github.com/van-namدينh/WORK.git`

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

### ...or create a new repository on the command line

```
echo "# WORK" >> README.md  
git init  
git add README.md  
git commit -m "first commit"  
git remote add origin https://github.com/van-namدينh/WORK.git  
git push -u origin master
```

### ...or push an existing repository from the command line

```
git remote add origin https://github.com/van-namدينh/WORK.git  
git push -u origin master
```

### ...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

>>>

#bsgmanycore project

link.ref4

[https://bitbucket.org/taylor-bsg/bsg\\_manycore/src/b3f179be012a?at=master](https://bitbucket.org/taylor-bsg/bsg_manycore/src/b3f179be012a?at=master)

try more

>>>

#more and more!

## # Shellscript and Linux command.

### **#useful shortcuts and keys**

```
ctrl+alt+t  
mkdir  
ls  
cd  
cp...  
exit  
>>>
```

### **# nice command.!**

I met/achieve it occasionally.!

```
cat inFile | grep pattern | sort > outFile
```

one more...

```
grep " FINISH \| I/O device \|^[Ee]rror" run.log
```

```
>>>
```

```
>>>
```

### **# zip, unzip file in linux**

!# use the commands:

```
zip -- help | head
```

```
tar -- help | head
```

example:

```
vnd@vnd:~/WORK/source-codes$ zip -r MatlabCode.zip MatlabCode  
adding: MatlabCode/ (stored 0%)  
adding: MatlabCode/day2.m (deflated 39%)  
adding: MatlabCode/hmcodes/ (stored 0%)  
adding: MatlabCode/hmcodes/h74.m (deflated 52%)  
adding: MatlabCode/day2-filterfdatool.m (deflated 18%)  
adding: MatlabCode/W1Dev1.m (deflated 46%)  
adding: MatlabCode/day2 (deflated 4%)  
adding: MatlabCode/exw1.m (deflated 46%)  
adding: MatlabCode/2output.png (deflated 21%)  
adding: MatlabCode/test2.m (deflated 59%)  
adding: MatlabCode/pdatoolgen.m (deflated 42%)  
adding: MatlabCode/ecg.mat (stored 0%)  
adding: MatlabCode/1output.png (deflated 11%)  
adding: MatlabCode/test.m (deflated 29%)  
adding: MatlabCode/untitled.m (deflated 42%)  
adding: MatlabCode/day2filterfdatool.m (deflated 35%)  
adding: MatlabCode/EXW1-results.zip (stored 0%)
```

this example create a zip file from an original folder MatlabCode.

```
>>>
```

### **#copy all files in a folder**

```
cp -R ~/.vim/header/ ~/Linux/vim/
```

Ghi chú:

**lệnh cp – R source\_folder destination\_folder.**

-R là tham số truyền để có thể copy tất các files trong thư mục nguồn

Link for ref.

<https://www.cyberciti.biz/faq/copy-command/>

(source link)

```
>>>
```

### **#Read file pdf commands**

```
evince noxim/doc/noxim_tutorial.pdf
```



or can also use

```
gnome-open noxim/doc/noxim_tutorial.pdf
```

>>>

### #Loops with BashScripts (BS)!

Can use this link.

<https://ryanstutorials.net/bash-scripting-tutorial/bash-loops.php>

>>>

### #Lệnh tìm "Key" trong thư mục, trong file (find something in linux).

!# Tham khảo hình sau:

```
find -name "*parity*"
vim ./RTL/Router_32_bit_credit_based_parity.vhd
cd ..
cd ..
*
find -name "*par*"
find -name "*parity*"
vim ./Router/credit_based/RTL/Router_32_bit_credit_based_parity.vhd
grep 'parity_checker_packet_detector' -r .
vim ./Router/credit_based/RTL/ParityChecker_packet_detector.vhd
```

>>>

### other way to find keyword in a folder

- cd to the folder.
- Use command: grep keyword \*.v

>>>

### # commands for checking the size of your working? (ref. from. Duy-Hieu Bui).

!# Reference from the following pic.

Để check dung lượng mình đang sử dụng, có thể dùng lệnh sau ở trong linux shell:  
du -hs ~/

Để phân tích các thư mục đang chiếm không gian lớn có thể dùng lệnh ncdu như sau:  
ncdu ~/

>>>

### # Lệnh tìm keys trong file bất kỳ

ví dụ:

```
grep -i error *.log
```

giải thích!

Câu lệnh trên, -i có nghĩa ko phân biệt chữ hoa hay thường; \*.log tức là tìm trong tất cả các file .log xem file nào có tồn tại chữ error mà không phân biệt chữ hoa hay thường, nghĩa là Error hay ErrOr đều được đưa ra kết quả hiển thị.

One nice command!

```
grep -ri rv32_opcodes.vh sistj/
```

```
grep -- help
```

chú ý: grep là tìm trong file, trong khi lệnh

```
find ../.. -iname "rv32_opcodes.vh"
```

là tìm tên file!

nice!

>>>

### #Most important command.

man command

! Trong cú pháp trên, ý nghĩa cần quan trọng nhất là khi bạn biết bất cứ từ khóa lệnh nào “command”, bạn có thể dùng cấu trúc này để kiểm tra xem cụ thể lệnh “command” đó là lệnh gì? chức năng gì? thực thi cấu trúc như thế nào?  
ví dụ: lệnh kiểm tra xem máy tính của bạn cài đặt hệ điều hành ubuntu version bao nhiêu? có thể dùng lệnh `lsb_release` là từ khóa nên có thể đánh là `man lsb_release`.  
Khi đó bạn sẽ được một màn hình giống như sau:

```
lsb_release(1)                                General Commands Manual                                lsb_release(1)

NAME
    lsb_release - print distribution-specific information

SYNOPSIS
    lsb_release [options]

DESCRIPTION
    The lsb_release command provides certain LSB (Linux Standard Base) and distribution-specific information.

    If no options are given, the -v option is assumed.

OPTIONS
    The program follows the usual GNU command line syntax, with long options starting with two dashes ('--'). A summary of options are included below.

    -v, --version
        Show the version of the LSB against which your current installation is compliant. The version is expressed as a colon separated list of LSB module descriptions.

    -i, --id
```

từ hình trên ta có thể dễ dàng thấy để kiểm tra version bao nhiêu chỉ cần gõ lệnh: `lsb_release -a` là sẽ được hình như sau:

```
vnd@vnd-ThinkPad-T420:~/c-c++programming$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 16.04.1 LTS
Release:        16.04
Codename:       xenial
```

good luck!

### # Your system is 32 or 64?

keys: `uname`

`uname -m`

(recommand for expanding: use `man uname`)

ex:

`uname -a`

!nice!)

>>>

### #Error during starting up with Ubuntu, keys: plymouth boot.

fig:

#### How do deactivate plymouth boot screen?



7



Lately, booting Ubuntu on my desktop has become seriously slow. We're talking two minutes. It used to take 10-20 seconds. Because of plymouth, I can't see what's going on. I would like to deactivate it, but not really uninstall it. What's the quickest way to do that? I'm using Precise, but I suspect a solution for 11.10 would work just as well.

### # Solution:

one way is followed figure

Easiest quick fix is to edit the grub line as you boot.

Hold down the shift key so you see the menu. Hit the e key to edit

Edit the 'linux' line, remove the 'quiet' and 'splash'

To disable it in the long run

Edit `/etc/default/grub`

Change the line – `GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"` to

```
GRUB_CMDLINE_LINUX_DEFAULT=""
```

And then update grub

```
sudo update-grub
```

### # Câu lệnh chèn thêm ký tự vào dòng bất kỳ trong file nguồn

#### !# Solution

```
paste -d "" number.txt src.txt >dst.txt
```

trong đó: **number.txt** là file số liệu cần thêm vào vị trí tương ứng trong file nguồn **src.txt** và lưu ra file output có tên là **dst.txt**.

### # Cập nhật font Times new roman trong Ubuntu.

!# Chúng ta có thể tham khảo lệnh sau:

```
sudo apt-get --reinstall install msttcorefonts
```

và lệnh:

```
sudo apt-get --reinstall install ttf-mscorefonts-installer
```

links tham khảo từ:

<https://askubuntu.com/questions/210680/installing-times-new-roman-font>

vnd!

>>>

### # Commands to find out process then kill it in Ubuntu?

Examples when you run vsim on ubuntu, you wanna kill it you can do as following step:

```
ps ax | grep vsim
```

kill -9 "number" - number = the number you can see after doing ps ax commands.

Ref.

```
vnd@vnd:~$ ps ax | grep vsim
5863 pts/2    S+        0:02 vish -- -vsim
5895 pts/20   S+        0:00 grep --color=auto vsim
vnd@vnd:~$ kill -9 5863
```

link ref:

<https://www.linux.com/learn/intro-to-linux/2017/5/how-kill-process-command-line>

an example for cm.!

```
ps aux | grep chrome
```

for more detail

>>>

```
ps aux | grep chrome
```

The *aux* options are as follows:

- a = show processes for all users
- u = display the process's user/owner
- x = also show processes not attached to a terminal

### #How to find any file with the key in linux?

!#

for example, I choose the file that includes the key is “Router”. I can use the command

```
find . -name Router*
```

after doing this, any file that includes the “Router...” will be shown

other ways can be referred from this link.ref4

<https://linode.com/docs/tools-reference/tools/find-files-in-linux-using-the-command-line/>

>>>

BR!

>>>

### #How to find in history commands (using grep)

# Tìm kiếm các từ khóa trong lịch sử lệnh (dùng grep)

```
history | grep svn | grep ref_flow
```

#! link tham khảo

[https://askubuntu.com/questions/55325/how-to-use-grep-command-to-find-text-including-subdirectories?](https://askubuntu.com/questions/55325/how-to-use-grep-command-to-find-text-including-subdirectories?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa)

[utm\\_medium=organic&utm\\_source=google\\_rich\\_qa&utm\\_campaign=google\\_rich\\_qa](https://www.cyberciti.biz/faq/debian-ubuntu-linux-grep-command-tutorial-examples/)  
link2

<https://www.cyberciti.biz/faq/debian-ubuntu-linux-grep-command-tutorial-examples/>

link 3:

>>>

### # Soft link commands in Linux

>>>

**Prob:** Khi biên dịch mà không trở đến được files hay đường dẫn không đúng, có thể dùng!  
ref from this picture

```
ln -sf ../../src/vcr_constants.v
ln -sf ../../src/whr_constants.v
ln -sf ../../src/rtr_constants.v
ln -sf ../../src/clib/c_constants.v
ln -sf ../../src/clib/c_functions.v
cd ..
make
history
hercules:~/WORK/source-codes/Open-Source-Network-on-Chip-Router-RTL/verif$
```

>>>

## #VIM SKILL et all

>>>>

### #Some basic note:

:set number = set number in vim

:wq = :x = write then quit.

:q! = quit without saving

>>>>

### # Kỹ thuật thay thế trong VIM

ví dụ áp dụng thêm comments vào đoạn code (dòng 109-119) của một file .hdl

bước 1: Nhấn tổ hợp phím ctrl + v (tại con trỏ dòng 109)

sau đó di chuyển con trỏ chuột đến dòng 119

bước 2: nhấn tổ hợp phím shift + i

bước 3: thêm dấu -- vào trước dòng 109.

bước 4: nhấn phím esc.

>>>>

### # search key “error” in a file (open by vim)?

!#

step 1: press ESC

step 2: type **:/error**

step 3: press keyboard N to the next finding. Then ESC to come back init. State.

### # keys to dev.

Basic vim github

▲ [Their Documentation says:](#)

#### 8 Cut and paste:

- ▼
1. Position the cursor where you want to begin cutting.
  2. Press **v** to select characters (or uppercase **V** to select whole lines).
  3. Move the cursor to the end of what you want to cut.
  4. Press **d** to cut (or **y** to copy).
  5. Move to where you would like to paste.
  6. Press **P** to paste before the cursor, or **p** to paste after.

Copy and paste is performed with the same steps except for step 4 where you would press y instead of d:

**d** = delete = cut

**y** = yank = copy

nice!

>>>>

### #Note for copying in VIM.

>>>>

basic:

note 1: convenient commands for copying codes.

I found the below command much more convenient. If you want to copy lines from 6 to 12 and paste from the current cursor position.

```
:6,12 co .
```

If you want to copy lines from 6 to 12 and paste from 100th line.

```
:6,12t100
```

Source: [https://www.reddit.com/r/vim/comments/8i6vbd/efficient\\_ways\\_of\\_copying\\_few\\_lines/](https://www.reddit.com/r/vim/comments/8i6vbd/efficient_ways_of_copying_few_lines/)

link: [https://www.reddit.com/r/vim/comments/8i6vbd/efficient\\_ways\\_of\\_copying\\_few\\_lines/](https://www.reddit.com/r/vim/comments/8i6vbd/efficient_ways_of_copying_few_lines/)

Note2: It should be tested and checked again!

You can do it as you do in vi, for example to yank lines from 3020 to the end, execute this command (write the block to a file):

```
:3020,$ w /tmp/yank
```

And to write this block in another line/file, go to the desired position and execute next command (insert file written before):

```
:r /tmp/yank
```

(Reminder: don't forget to remove file: /tmp/yank)

>>>

!#

we can ref some notes from this:

[http://vim.wikia.com/wiki/Copy, cut and paste](http://vim.wikia.com/wiki/Copy,_cut_and_paste)

however, one key is:

copy 1 word in vim we use `y1` then go to the position where you wanna paste it and press `p`

copy n word in vim we use `yn` then go to the position then press `p`.

copy n word in m time we use `yn` then go to the position then press `m p`

>>> enjoy it!

>>>

**# useful commands more and more!**

>>>

>>>

**#other text editor that I can use for programming.**

Keys: Sublime Text

after install in ubuntu, we can open the command line and type

`subl&`

>>>

>>>

## **# CODEFIGHTS and GNU Make (Bash Script)**

>>>

### **#information**

>>>

username: van-namdinh

link: <https://codefights.com/>

email: [vndprogrammings@gmail.com](mailto:vndprogrammings@gmail.com)

password: normal\_newnum

>>>

other link.

<https://www.hackerrank.com/onboarding/challenge>

>>>

### **GNU Make.**

>>>

# Link ref1.

[https://www.gnu.org/software/make/manual/html\\_node/index.html#SEC\\_Contents](https://www.gnu.org/software/make/manual/html_node/index.html#SEC_Contents)

>>>

>>>

#PYTHON

>>>

**#very useful link to practice!**

>>>

Link 1:

<https://www.learnpython.org/>

>>>

>>>

Link 2:

<https://campus.datacamp.com/courses/intro-to-python-for-data-science/>

>>>

>>>



## ENGLISH

>>>

### #Useful link and ref for E. study.

>>>

>>>

#### **#one of nice link.ref4**

>>>

<https://www.rong-chang.com/>

>>>

other:

>>>

<https://www.eslfast.com/>

>>>

#### **#For listening skills**

>>>

[https://money.cnn.com/2018/07/20/pf/make-more-money-at-work/index.html?utm\\_source=quora&utm\\_medium=referral](https://money.cnn.com/2018/07/20/pf/make-more-money-at-work/index.html?utm_source=quora&utm_medium=referral)

>>>

choose the most like one to listen to

<https://money.cnn.com/>

>>>

#### **# A nice link that I should care!**

>>>

<http://education.yourdictionary.com/>

>>>

#### **#activities for Lesson-Plan in the class.**

>>>

<http://esl.yourdictionary.com/lesson-plans/>

>>>

others:

>>>

#Lộ trình học TA cho người mới bắt đầu

>>>

#### **#Web tiếng anh 123 của người việt**

Username: english\_study\_fun

Pass: Tienganh123\_newnum

Email: [hocngoingu2586@gmail.com](mailto:hocngoingu2586@gmail.com)

Nạp và kích hoạt thẻ VIP ngày **16/6/2017**

Hạn một năm sử dụng đến **15/6/2018**.

>>>

Username: esvnd2586

pass: Nguyenxxxthuyan\_number(2586) => onlynumber\_171117

email: [vndprogrammings2586@gmail.com](mailto:vndprogrammings2586@gmail.com)

>>>

#### **#Grammar checking**

links:

<https://app.grammarly.com/>

email: [namdv2586@gmail.com](mailto:namdv2586@gmail.com)

password: normar\_oldnum

>>>

**#Link download from FB**

<https://en.savefrom.net/9-how-to-download-facebook-video.html>

>>>

**#everyday English review and study should be checked here**

<http://www.tolearnenglish.com/index.php>

username: vnd2586

pass: normal\_withnum.

>>>

**#website to improve writing skill**

<https://writeandimprove.com/workbooks#/wi-workbooks>

email ĐK: [hocngoangu2586@gmail.com](mailto:hocngoangu2586@gmail.com)

pass: Dinhvannam2586

>>>

**#bbc ref link to study English**

**!# English study with BBC!**

<http://www.bbc.co.uk/learningenglish/>

username or email: [hocngoangu2586@gmail.com](mailto:hocngoangu2586@gmail.com)

pass: normal\_new\_number

>>>

<http://www.bbc.co.uk/worldservice/learningenglish/teach/>

>>>

**#Learn English from voa channels**

link:

<https://learningenglish.voanews.com/p/5644.html>

>>>

>>>

**# for beginner to listen and repeat!**

nice!

<https://www.rong-chang.com/supereasy/>

>>>

**#Một số link hỗ trợ luyện đọc cho ielts (lấy từ tuần 5 của khóa học Understanding IELTS)**

#link 1

[http://learnenglish.britishcouncil.org/en/stories-poems?](http://learnenglish.britishcouncil.org/en/stories-poems?utm_source=IETLS7&utm_campaign=FL_MOOC_Courses&utm_medium=content_wk5)

[utm\\_source=IETLS7&utm\\_campaign=FL\\_MOOC\\_Courses&utm\\_medium=content\\_wk5](http://learnenglish.britishcouncil.org/en/stories-poems?utm_source=IETLS7&utm_campaign=FL_MOOC_Courses&utm_medium=content_wk5)

#link 2

[http://learnenglishteens.britishcouncil.org/uk-now/literature-uk?](http://learnenglishteens.britishcouncil.org/uk-now/literature-uk?utm_source=IETLS7&utm_campaign=FL_MOOC_Courses&utm_medium=content_wk5)

[utm\\_source=IETLS7&utm\\_campaign=FL\\_MOOC\\_Courses&utm\\_medium=content\\_wk5](http://learnenglishteens.britishcouncil.org/uk-now/literature-uk?utm_source=IETLS7&utm_campaign=FL_MOOC_Courses&utm_medium=content_wk5)

#link 3

[http://learnenglishteens.britishcouncil.org/study-break/graded-reading?](http://learnenglishteens.britishcouncil.org/study-break/graded-reading?utm_source=IETLS7&utm_campaign=FL_MOOC_Courses&utm_medium=content_wk5)

[utm\\_source=IETLS7&utm\\_campaign=FL\\_MOOC\\_Courses&utm\\_medium=content\\_wk5](http://learnenglishteens.britishcouncil.org/study-break/graded-reading?utm_source=IETLS7&utm_campaign=FL_MOOC_Courses&utm_medium=content_wk5)

username: Van-Nam DINH

email: [hocngoangu2586@gmail.com](mailto:hocngoangu2586@gmail.com)

pass: Dixxx\*\*\*@123.

>>>

**# English for kids**

<http://learnenglishkids.britishcouncil.org/en>

username: Van-Nam DINH

email: [hocngoangu2586@gmail.com](mailto:hocngoangu2586@gmail.com)

pass: Dixx\*\*\*@123

>>>

### **#more for kids**

<http://www.freddiesville.com/lesson-ville/>

>>>

### **#English for kids**

#link 1

<http://www.kidsworldfun.com/>

#link 2:

>>>

### **#learn English**

<https://learnenglish.britishcouncil.org/en>

username and password the same.

>>>

### **#Teaching English**

<https://www.teachingenglish.org.uk/>

username: English\_study

pass: same!

>>>

### **#should check this link**

<http://www.cambridgeenglish.org/teaching-english/>

>>>

### **#very useful from A-Z I think!**

<http://www.esl-lab.com/>

to download should use the following link

<https://video-download.online/site/esl-lab.com>

>>>

### **#some courses from BC (British Council)**

<https://www.britishcouncil.vn/day-tieng-anh/khoa-hoc>

<http://www.teachingenglish.org.uk/>

<http://www.teachingenglish.org.uk/getting-started-resources>

>>>

### **#Very interesting link:**

<https://learnenglish.britishcouncil.org/en>

email: [hocngoangu2586@gmail.com](mailto:hocngoangu2586@gmail.com)

username: Van-Nam DINH

Pass: normal\_number\_@123Z

### **Others:**

<http://learnenglish.britishcouncil.org/en/content>

>>>

>>>

### **#For teaching in conversation classes**

<http://www.englishcurrent.com/teachers/teach-first-english-conversation-class/>

>>>

### **#amazing forum and notes for studying English free!**

Link for help (how to use paltalk)

<http://viet-esl.info/showthread.php?tid=1668&highlight=paltalk>

Home page

<http://viet-esl.info/>

username: van-nam

email for registering: [hocngoangu2586@gmail.com](mailto:hocngoangu2586@gmail.com)

pass: 7qrCjH95

newone:

>>>

### #Dictionary translate ref.

Link 1: oxford

<https://en.oxforddictionaries.com/>

link 2: Cambridge

<http://dictionary.cambridge.org/>

link 3: gg translate

<https://translate.google.com/>

>>>

### #to learn vocabulary

<https://www.memrise.com/home/>

user name: van-nam

pass: normal\_num.

email: [hocngoangu2586@gmail.com](mailto:hocngoangu2586@gmail.com)

>>>

### TOEIC

# Link for studying TOEIC

<https://www.toeicmoingay.com/>

email: [hocngoangu2586@gmail.com](mailto:hocngoangu2586@gmail.com)

username: toeicforme

pass: 2586@123vnd\_normal.

#some useful info to know about TOEIC

<http://cfl.edu.vn/toeic/thang-diem-va-cach-tinh-diem-bai-thi-toeic/>

>>>

>>>

### IELTS

>>>

### # IELTS testing online (practicing English is perfect)

<http://ieltsonline-tests.com/>

>>>

### #IELTS daily life! (From Wednesday, May 31, 2017).

Wednesday, May 31, 2017.

### #Cam.10.1.Speaking

P1. Ex: Weekends

- How do you usually spend your weekends [why?].

A1: Well, I often visit my hometown at the weekends. Because as you know, I am so busy in the week days, so I try to visit my family at the weekends always.

- Which is your favorite (favourite) part of the weekend? [Why?].

A1: Well, when I visit my parents at the weekends, the best time almost in the evening. Because, my parents are farmers, so they are still working at the weekends. We just can have more time to do some activities together in the evening, such as eat, share weekday's stories, jobs, experiences, and with me, it is really meaning time for us. It brings us closer.

- Do you think your weekends are long enough? [Why/WN?].

A1: Well, I think it is enough for me, but it would be better if I have extra time at the weekends. Because, as you know, my father have 11 siblings, so I could not visit them all at the weekends. If there were extra time, I would like to visit them all.

- How important do you think it is to have free time at the weekends? [Why?].

A1: It will be so bad if there are not free time at the weekends, at least with me! For me, working hard all week days are so tired. So, I balance myself by practicing gym every day and visiting my

family at the weekends. I could not imagine what happens to me if there is no free time at the weekends.

P2. Describe someone you know who does something well. You should say:

- Who this person is
- How you know this person
- What they do well
- And explain why you think this person is so good at doing this.

P3. Discussion topics

Salaries and abilities:

- What skills and abilities do people most want to have today? Why?
- Which skills should children learn at school? Are there any skills which they should learn at home? What are they?
- Which skills do you think will be important in the future? Why?

Salaries for skilled people:

- Which kinds of jobs have the highest salaries in your country? Why is this?
- Are there any other jobs that you think should have high salaries? Why do you think that?
- Some people say it would be better for society if everyone got the same salary. What do you think about that? Why?

#Cam.10.1.Writing.

Task 1: pie chart.

Task 2: Present a written argument or case to an educated reader with no specialist knowledge of the following topic.

**It is important for children to learn the difference between right and wrong at an early age. Punishment is necessary to help them learn this distinction.**

**To what extent do you agree or disagree with this opinion?**

**What sort of punishment should parents and teachers be allowed to use to teach good behavior to children?**

Give reasons for your answer and include any relevant examples from your own knowledge or experience.

**Sample essay:**

>>>>

**#Tips for IELTS.**

>>>>

#IELTS materials

<http://ieltsmaterial.com/top-8-best-ielts-preparation-books/>

>>>>

**# IELTS target 6.5 from IELTS fighter**

>>>>

[http://ielts-fighter.com/tin-tuc/Lo-trinh-luyen-thi-IELTS-online-free-level-6-5\\_mt1464170289.html](http://ielts-fighter.com/tin-tuc/Lo-trinh-luyen-thi-IELTS-online-free-level-6-5_mt1464170289.html)

>>>>

>>>>

**#General Tips:**

>>>>

### **#Listening Tips:**

>>>>

### **#Reading Tips:**

>>>>

### **#Writing Tips:**

>>>>

### **#Speaking Tips:**

>>>>

## **#Series IELTS writing topic samples of band 7 from Blog IELTS**

### **Essay 1: popular events**

Popular events like the football world cup and other international sporting occasions are essential in easing international tensions and releasing patriotic emotions in a safe way. To what extent do you agree or disagree with this opinion?

### **Essay 2: Internet connecting people**

Some people say that the Internet is making the world smaller by bringing people together. To what extent do you agree that the internet is making it easier for people to communicate with one another?

### **Essay 3: Space exploration**

With all the problems in the world today, spending money on space exploration is a complete waste. The money could be better spent on other causes.

### **Essay 4: Children and rules**

In some countries children have very strict rules of behavior, in other countries they are allowed to do almost anything they like. To what extent should children have to follow rules?

### **Essay 5: Smoking in public places**

Some businesses prohibit smoking in any of their offices. Some governments have banned smoking in all public places. Do you agree or disagree that this is the right course of action? Give reasons for your opinion.

### **Essay 6: The Internet as a source of information**

The Internet when used as a source of information, has more drawbacks than advantages. To what extent do you agree with this statement?

### **Essay 7: Computers in our life**

We are becoming increasingly dependent on computers. They are used in business, hospitals, crime detection and even to fly planes. What things will they be used for in the future? Is this dependence on computers a good thing or should we be more suspicious of their benefits?

### **Essay 8: Advertising**

Advertising is all around us, it is an unavoidable part of everyone's life. Some people say that advertising is a positive part of our lives while others say it is a negative one. Discuss both views and include your own opinion. Give reasons for your answer and include any relevant examples from your own knowledge or experience. You should write at least 250 words.

### **Essay 9: Should wealthy nation share their wealth?**

Should wealthy nations be required to share their wealth with poorer nations by providing them with things such as food and education? Or is this the responsibility of the governments of poorer nations to look after their citizens?

### **Essay 10: Should Tobacco be prohibited?**

Should the same laws which prohibit the sale and consumption of heroin be applied to tobacco? Discuss.

### **Essay 11: Capital punishment**

Without capital punishment (the death penalty) our lives are less secure and crimes of violence increase. Capital punishment is essential to control violence in society. To what extent do you agree or disagree?

### **Essay 12: Environmental problems**

Nowadays environmental problems are too big to be managed by individual persons or individual countries. In other words, it is an international problem. To what extent do you agree or disagree?

### **Essay 13: News on TV**

News editors decide what to broadcast on television and what to print in a newspaper. What factors do you think influence their decisions? Do we become used to bad news? Would it be better if more good news was reported?

### **Essay 14: The Internet and communication**

Some say that the Internet is making the world smaller by bringing people together. To what extent do you agree that the Internet is making it easier for people to communicate with one another?

### **Essay 15: Working children**

In many countries children are engaged in some kind of paid work. Some people regard this as completely wrong, while others consider it a valuable work experience, which is important for learning and taking responsibility. What is your opinion?

### **Essay 16: Computer in the future**

We are becoming increasingly dependent on computers. They are used in business, crime detection and even to fly planes. What things will they be used for in future? Is this dependence on computers a good thing or should we be more suspicious of their benefits?

### **Essay 17: Should students travel?**

Some people think that students who don't take a break in studies between the high school and the university are at disadvantage compared to students who travel and work after high school before further continuing their education. Do you agree or disagree?

>>>>

### **#IELT LISTENING**

#### **#useful link for practicing**

<http://www.ieltsclick.com/p/listening.html>

>>>>

### **#IELTS SPEAKING**

<http://www.westudy.vn/>

user name: Van-Nam Dinh

email: [hocnguoaingu2586@gmail.com](mailto:hocnguoaingu2586@gmail.com)

pass: normal@\_number

>>>>

#### **#Useful link for speaking tasks**

<http://www.ieltsbuddy.com/ielts-speaking-questions.html>

>>>>

### **# Questions and answers related to IELTS and English study. (QUORA.com)**

#### **#nice1: it should be noted later after reading**

<https://www.quora.com/How-do-I-improve-my-English-40>

>>>>

Link 1:

<https://www.quora.com/What-are-the-best-resources-for-IELTS-preparation>

Link 2:

<https://www.quora.com/What-is-the-best-way-to-prepare-for-IELTS-1>

>>>>

### **# Useful links for IELTS studying.**

Link 1:

<http://ieltspackage.blogspot.com/>

Notes for reading skill:

<http://ieltspackage.blogspot.com/2013/07/time-management-in-ielts-reading.html>

>>>>

Link 2:

<http://www.ieltsclick.com/>

>>>

### #From British Council

<https://www.britishcouncil.vn/en/exam/ielts/courses-resources/videos>

>>>

### # Notes for IELTS Writing

(ref. from p9/172 of the book: Academic writing practice for IELTS – Sam McCarter).

#### ✓ Word length and timing

For task 1: 150 ÷ 180 words should be done in 20 minutes.

For task 2: 250 ÷ 300 words should be done in 40 minutes.

#### ✓ 4 different criteria for assessing your writing task 1: (Band 7.0).

- **TA: Task Achievement:** information, overview, detail and accuracy-based.
  - Cover requirement of the task.
  - Presents a clear overview of main trends, differences or stages.
  - Clearly presents and highlights key features/bullet points but could be more fully extended.
- **CC: Coherence and Cohesion:** Organization of information, paragraphing, and use of linking devices-based.
  - Logically organizes information and ideas; there is clear progression throughout.
  - Uses a range of cohesive devices appropriately.
- **LR: Lexical Resource:** Use of vocabulary, spelling and the number of mistakes-based.
  - Use the sufficient range of vocabulary to allow some flexibility and precision.
  - Use less common lexical items with some awareness of style and collocation.
  - May produce occasional errors in word choice, spelling and/or word formation.
- **GRA: Grammatical Range and Accuracy:** Base on sentence structures, control of grammar and number of mistakes.
  - Use a variety of complex structures.
  - Produces frequent error-free sentences.
  - Has good control of grammar and punctuation but may make a few errors.

✓ Etc.

>>>

### # Keys for developing IELTS practicing

- ✓ Idioms for IELTS 7+.
- ✓ Structure for IELTS 7+.
- ✓ Writing IELTS 7+.
- ✓ Speaking IELTS 7+.

All these keys should be up-to-dated little by little by VND.

>>>

### #IELTS speaking practices

#01: What are your strong points? What is your weakness?

SA1: No man is perfect in this world and similarly I have certainly positive and negative aspects as well. Talking about the positive aspects. I am a very hardworking person and believe in hard work and persistence. I believe that with hard work everything is achievable. Secondly, I am a positive person who thinks positively. Besides that I am an honest person who believes in honesty because



without honesty nothing can survive for a long time. I am a good listener, too. And hence I absorb many things from others.

As far as negative aspects are concerned, I trust people very fast and this sometimes creates problems in my life. I am also very sensitive to nature and this creates a lot of tension in my life. I am also quite a cool person and I don't get angry fast. Sometimes I feel people take advantage of this. When I am involved in my work I cannot think of anything else, sometimes I don't remember important things going on in the family. I know that this is not good, and so I am trying to rectify the same.

#02:

### #Link for IELTS speaking cue card

<http://ieltsband7.com/category/ielts-preparation/cue-card/>

>>>

### #Free practice tests for IELTS

<http://takeielts.britishcouncil.org/prepare-test/free-practice-tests>

>>>

### #Phrasal Verbs and Nice sentences

>>>

The power of your **subconscious mind** = Sức mạnh tiềm thức.

### #Vocabularies + notes + links et al

>>>

>>>

### #Table of new words day by day.

>>>

>>>

Date	Vocabulary	Meaning	Exam	Syn. & Ant.
30May2018				
	Parlance	lời nói, trò chuyện		

>>>

>>>

No.	Vocab.	Meaning	Examples	Syn. & Ant.
N				

>>>

### #Oxford dictionaries... (english)

<https://en.oxforddictionaries.com/>

>>>

### #Cambridge dictionaries (english)

<http://dictionary.cambridge.org/>

>>>

# Date on 12 September 2017...

Here are some explanations, I hope they help.

Q 1. Ascent=go up, rise.

Q 2. Birth=being born.

Q 3. Compliment=praise.

Q 4. Cord=thick string.

Q 11. Bait=to attract, to tempt.

Q 12. Genius=a person with a great IQ.

Q 14. Hypocritical=hypocrite, insincere.

From this link

<https://www.tolearnenglish.com/cgi2/myexam/voir2r.php?id=116772>

>>>

## Computer and Architecture/ Organization

>>>

**#very basic about flip-flop (related to latch and gates)!**

<http://www.electronicsteacher.com/computer-architectures/digital-circuits/flip-flops.php>

! Note for more understanding about FF, Latch and gate... in computer architecture.

>>>

**#What is “cache controller” ?**

>>>

**Link1: detail of cache controller:**

>>>

<http://www.cl.cam.ac.uk/research/srg/bluebook/32/mdh9/node6.html>

>>>

**# Gem5, Garnet2.0 et.al**

>>>

**# Link for tutorial and learning gem5**

>>>

<http://learning.gem5.org/book/index.html>

>>>

**# Link wiki main\_page of Gem5**

>>>

[http://www.gem5.org/Main\\_Page](http://www.gem5.org/Main_Page)

>>>

**# Link for teaching Garnet2.0 with Gem5**

>>>

[http://tusharkrishna.ece.gatech.edu/teaching/garnet\\_gt/](http://tusharkrishna.ece.gatech.edu/teaching/garnet_gt/)

>>>

**# Interconnection Network Course (2017)**

>>>

[http://tusharkrishna.ece.gatech.edu/teaching/icn\\_s17/](http://tusharkrishna.ece.gatech.edu/teaching/icn_s17/)

>>>

**# Garnet2.0 with gem5 on wiki**

>>>

<http://www.gem5.org/Garnet2.0>

>>>

**# Link for up2date of Garnet2.0 dev.**

>>>

<http://synergy.ece.gatech.edu/tools/garnet/>

#

>>>

## # NoC: Network-on-Chip et al.

>>>

### # Courses online

There are many resources and useful keys according to the following links!

#course 1

[http://tusharkrishna.ece.gatech.edu/teaching/icn\\_s17/](http://tusharkrishna.ece.gatech.edu/teaching/icn_s17/)

#course 2

[http://tusharkrishna.ece.gatech.edu/teaching/icn\\_s18/](http://tusharkrishna.ece.gatech.edu/teaching/icn_s18/)

>>>

### #Keys and notes:

!#

Fault-tolerant architecture

fault-tolerant algorithm

injection fault techniques

Transient faults; permanent faults; intermittent faults

HW injection faults, SW injection Faults, injection faults simulation...

>>>

### #Các khái niệm cơ bản cần lưu ý

>>>

>>>

### #Simulator and blog dev.

>>>

<https://networkonchip.wordpress.com/2015/11/02/what-is-the-most-popular-full-system-simulator/>

>>>

>>>

### #Ref. and related to my PHD courses.

Keys: gem5, garnet2.0; fault tolerance model, hotSpot, ORION; etc.

### #Fault model tools for NoC

<http://www.princeton.edu/~carch/kaisopos/FaultModel/>

more:

<http://www.princeton.edu/~carch/kaisopos/FaultModel/GarnetIntegration.htm>

more and more....

Link update for HotSpot

<http://lava.cs.virginia.edu/HotSpot/>

more...

>>>

### #Gem5 home page

[http://www.gem5.org/Main\\_Page](http://www.gem5.org/Main_Page)

=> from this site, can search with garnet2.0 key word and then expand it as much as possible.

>>>

### # Deadlock-free for reference links.

!# we can read the following link to know more about definition of deadlock-free.

<https://cwiki.apache.org/confluence/display/ASTERIXDB/Deadlock-Free+Locking+Protocol>

>>>

>>>

### #Soft errors and related things

#Link to understand more about soft-errors (causes by)?

<https://www.slideshare.net/iROCTech/i-roc-webinar-do-i-really-need-to-worry-about-soft-error-may302013>

nice! It is a nice link that includes some definitions about soft-errors.

>>>

## #Basic:

(from william James' book)

### 1.3 Network Basics

To meet the performance specifications of a particular application, such as those described above, the network designer must work within technology constraints to implement the *topology*, *routing*, and *flow control* of the network. As we have said in the previous sections, a key to the efficiency of interconnection networks comes from the fact that communication resources are shared. Instead of creating a dedicated channel between each terminal pair, the interconnection network is implemented with a collection of shared router nodes connected by shared channels. The connection pattern of these nodes defines the network's *topology*. A message is then delivered between terminals by making several *hops* across the shared channels and nodes from its source terminal to its destination terminal. A good topology exploits the properties of the network's packaging technology, such as the number of pins on a chip's package or the number of cables that can be connected between separate cabinets, to maximize the bandwidth of the network.

Once a topology has been chosen, there can be many possible paths (sequences of nodes and channels) that a message could take through the network to reach its destination. *Routing* determines which of these possible paths a message actually takes. A good choice of paths minimizes their length, usually measured as the number of nodes or channels visited, while balancing the demand placed on the shared resources of the network. The length of a path obviously influences latency of a message through the network, and the demand or *load* on a resource is a measure of how often that resource is being utilized. If one resource becomes over-utilized while another sits idle, known as a load imbalance, the total bandwidth of messages being delivered by the network is reduced.

*Flow control* dictates which messages get access to particular network resources over time. This influence of flow control becomes more critical as the utilization of resource increases and good flow control forwards packets with minimum delay and avoids idling resources under high loads.

=> translating!...

>>>>

#### #Nhiệm vụ chính của PHD course theo topic;

Dưới đây là thứ tự nhiệm vụ nghiên cứu cần cố gắng hiểu bản chất bên tiếng việt, sau đó khi viết bài luận có thể viết bằng tiếng việt hay tiếng anh phụ thuộc vào thời điểm lúc đó.

Topic 1: The NoC architecture including the topology, communication mechanism, routing algorithm, flow control and so on

**Chủ đề 1: Kiến trúc mạng trên chip gồm topology, cơ chế truyền thông, các giải thuật định tuyến, điều khiển luồng và một vài vấn đề khác.**

Vấn đề 1: **topology**:

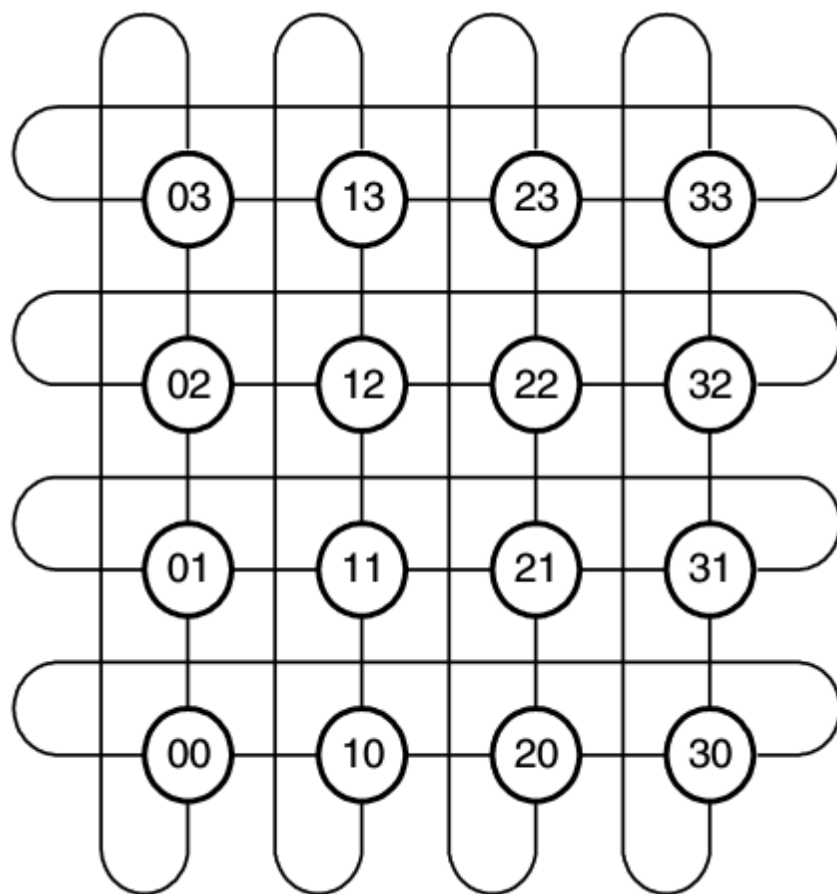
### 1.3.1 Topology

Interconnection networks are composed of a set of shared router nodes and channels, and the *topology* of the network refers to the arrangement of these nodes and channels. The topology of an interconnection network is analogous to a roadmap. The channels (like roads) carry packets (like cars) from one router node (intersection) to another. For example, the network shown in Figure 1.6 consists of 16 nodes, each of which is connected to 8 channels, 1 to each neighbor and 1 from each neighbor. This particular network has a *torus* topology. In the figure, the nodes are denoted by circles and each pair of channels, one in each direction, is denoted by a line joining two nodes. This topology is also a *direct network*, where a terminal is associated with each of the 16 nodes of the topology.

ref from William's book about Interconnection Network. (Page 13; 40 of 581).

Các mạng liên kết với nhau bao gồm tập hợp các nút định tuyến và các kênh, và topology của mạng chính là sự sắp xếp của những nút và các kênh này. Topology của mạng liên kết tương tự như một roadmap (lộ trình/bản đồ đường nối) vậy. các kênh (channels) giống như các con đường, và mạng các gói tin (giống như các chiếc ô tô) từ các nút định tuyến(intersection-ngã tư??) đến các nút định tuyến khác.

Figure 1.6



Cái hình củ chuỗi này tham khảo nên mờ, có time draw lại sau! :(... Một topology tốt khai thác được các đặc trưng của các công nghệ chuyển mạch gói sẵn có để đạt được các yêu cầu về độ trễ và băng thông của các ứng dụng với giá thành thấp nhất. Để tối ưu băng thông, một topology nên saturate (thỏa mãn?) the băng thông hai chiều (bisection bandwidth), băng thông qua midpoint của hệ thống, cung cấp bởi các công nghệ chuyển mạch gói. ??? Vấn đề chốt ở đây là **công nghệ**

**chuyển mạch gói là gì? có những công nghệ chuyển mạch gói nào hiện nay đang hỗ trợ cho các topology? và cái nào đã được áp dụng cũng như đang nghiên cứu, so sánh....**

**d!**

**>>>**

## # ARDUINO

>>> (should be updated at least 5M each day)! In the evening (before going to sleep).

#Web học arduino từ A-Z (VNese).

Link:

<http://arduino.vn/>

user: Van-Nam Dinh

email: [vndprogrammings@gmail.com](mailto:vndprogrammings@gmail.com)

pass: normal\_number@123A

notes: should be accepted and read more and share to get more knowledge!

>>>



## # Digital Design (Essential basic knowledge)

>>>

### **# interesting course from MIT**

<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-111-introductory-digital-systems-laboratory-spring-2006/>

should follow it!

>>>

### **#Link reference for vlsi theory.... (lecture)**

>>>

<http://pages.hmc.edu/harris/cmosvlsi/4e/lect/>

Link for slides etc...

<http://pages.hmc.edu/harris/class/e158/>

>>>

>>>

### **#Essential Knowledge: based on Dolous Documents keys**

#Keys for introduction: ASICs, Synchronous design techniques, Design Processes HDLs

>>>

### **# Comprehensive knowledge about Digital design (FSM, CL, Sequential Logic)**

!#

Reading books

- item 7.3 “Clocked Synchronous State-Machine Analysis” in Digital Design (John F. Wakerly).

Summary!:

try to answer some following questions:

- Where we can use FSM?
- When we need to use FSM?
- How does a FSM work?
- ...

still up2date!

>>>

### **#Link ref. for Digital Design (so great!)**

!#

Should read the book Digital Design of Wakerly then read more the following

<http://wakerly.org/DDPP/DDPP4student/>

it is really well.

>>>

### **# tool need to know**

DVT nice!

<https://dvteclipse.com/products/dvt-eclipse-ide>

>>>

### **# full adder vs half adder**

>>>

[https://www.electronics-tutorials.ws/combinational/comb\\_7.html](https://www.electronics-tutorials.ws/combinational/comb_7.html)

>>>

### **#Elaborate in DC (synopsys tool) ASIC world**

!#

We can ref. from. This link

<https://forums.xilinx.com/t5/Synthesis/what-exactly-is-elaborating-a-design/td-p/682043>

according to this link, I can get the following answer that I can get many interesting things!

Thank to the beautiful life!

## Re: what exactly is 'elaborating' a design? [ Edited ]

Options ▾

02-17-2016 05:42 AM - edited 02-17-2016 06:47 AM

When we talk about "implementation", we understand that we are talking about a number of separate processes:

- opt\_design
- place\_design
- route\_design
- (and a number of others)

In the tool, these are split out a separate processes, each with their own options.

From the user point of view, "synthesis" is a single process, but, in reality, it is a number of processes

- elaborate the design
- apply constraints to the design
- do high level optimizations of the design
- technology map the design
- do low level optimizations of the design

The first step, elaboration, is reading in your RTL file (which is text) and recognizing bits of code that represent real hardware structures. Once recognized, these are converted (in Vivado synthesis case) into "generic technology cells" - abstract things like registers, adders, comparators, multiplexers, arbitrarily wide gates, etc...

This step is necessary since the next steps (high and low level optimizations) are timing driven, and hence need constraints. But, constraints cannot be applied to your RTL (which is text) - they need to be applied to a netlist. So elaboration creates the netlist of generic technology cells.

In normal flows, we simply do elaboration as part of synthesis - its more or less invisible (although it may show up in the synthesis log). However, we can ask Vivado synthesis to "stop" after elaboration, and give us access to the netlist of generic technology cells.

There are only a couple of reasons to do this:

- 1) to make sure that the synthesis tool understood our RTL code. We can look at things like hierarchical structure, check for things like unconnected inputs and outputs, make sure it recognized higher level structures (like RAMs), all before it goes ahead with the rest of the steps
- 2) to verify and create constraints. The next steps of synthesis (optimization) need constraints. These constraints, though, need to be correct for this "generic technology" representation of your design. Generally, the constraints that you write for your design will end up being applicable for both the generic tech representation and the technology mapped representation (where the design is converted to Xilinx Basic Elements or BELs). Sometimes, though, the structure and naming of the generic technology netlist means that the constraints need to be ever so slightly different (or written differently so that they can apply to both forms of netlist). By opening the elaborated design, you can experiment with constraints to find the format that accomplishes what you want - these can then be used for synthesis or for both synthesis and implementation

Once you have done all this, you close the elaborated design, and it goes away (i.e. you cannot "continue" the rest of the synthesis process). When you synthesize the design, it will re-elaborate it as part of the synthesis run.

By the way, the elaboration process is actually done by the synthesis process - in non-project batch mode it is done with the command

```
synth_design -rtl
```

This specifically tells the synth\_design process to stop after the elaboration has been completed.

Avrum

>>>

### one more answer

I'd add a few more notes to avrum's answer. I break that first synthesis step down (mentally) into 2 steps:

Compilation of code  
Elaboration of code

During compilation, the tool's looking at basic syntax, and verifying there's no syntax errors. Verifying that you're code obeys the rules of the language, pretty much on just a file-by-file basis.

Next, during elaboration, the tool's checking more global, connectivity, and other rules to the entire design.

During elaboration, it's evaluating parameters (generics), setting port sizes, etc, accordingly. The tool's checking to make sure the connectivity is legit. i.e. a top level module calls a sub-module with a port "A". Well does port "A" actually exist on the sub-module? Is it the correct width? And many similar checks. After elaboration, it's ready for the next steps as avrum's described.

For me this has its roots in Cadence's simulation software. You used to call three separate tools to simulate:

```
ncverilog  
ncelab  
ncsim
```

So the steps there were very explicit. Cadence probably didn't invent the terms, but that's where I became familiar with them.

Modelsim's similar - where you run:

```
vlog  
vsim
```

Here the vsim command does the elaboration, and simulation. Actually, for Mentor, there's some flows which include a "vopt" command after vlog, which (among other things) does an elaboration.

Regards,

Mark

it's nice!

>>>



## #ICTU guide 2 Students menu

>>>>

### # Đề tài thực tập tốt nghiệp

>>>>

#01: Tìm hiểu về các công cụ mô phỏng mạng trên chip mã nguồn mở.  
(time: 01/02/2018 to 15/04/2018)

#### Mục tiêu:

- Tìm hiểu tổng quan về hệ thống mạng trên chip.
- Các công cụ mã nguồn mở hỗ trợ mô phỏng mạng trên chip.
- Làm quen dần với môi trường làm việc trên Linux.

#### Nội dung chính:

- Các khái niệm cơ bản về mạng trên chip.
- Giới thiệu về các mã nguồn mở hỗ trợ mô phỏng mạng trên chip.

**Tóm tắt đề tài: wait for discussing with student.**

Students' info:

- full name: Nguyễn Thị My
- ID no.: DTC145D5103020020; Điện tử ứng dụng K13A.

#02: Tìm hiểu về mô hình mạng trên chip Garnet2.0.  
(time: 01/02/2018 to 15/04/2018)

#### Mục tiêu:

- Tìm hiểu tổng quan về hệ thống mạng trên chip.
- Cấu trúc mạng trên chip Garnet2.0.
- Làm quen dần với môi trường làm việc trên Linux.

#### Nội dung chính:

- Tổng quan về mạng trên chip.
- Giới thiệu về hệ thống Garnet2.0.

**Tóm tắt đề tài: wait for discussing with student.**

Students' info:

- full name: Hoàng Anh Oanh.
- ID no.: DTC145D5103010101; Điện tử ứng dụng K13A.

### # Đề tài đồ án tốt nghiệp.

>>>>

#01: Thực thi đánh giá độ ảnh hưởng của tốc độ tải tin lên độ trễ truyền thông trung bình gói tin trong hệ thống mạng trên chip dựa trên nền tảng Gem5.

(time: 01/02/2018 to 27/05/2018)

#### Mục tiêu:

- Thực thi mô phỏng một NoC dựa trên nền tảng Gem5.
- Khảo sát sự ảnh hưởng của tốc độ tải tin vào mạng, để đánh giá hiệu năng mạng.
- Làm quen với nền tảng mã nguồn mở trên Linux (Ubuntu) (cái này Students đã pt).

#### Nội dung chính:

- Chương 1: Giới thiệu về hệ thống mạng trên chip.
- Chương 2: Cơ sở lý thuyết và nền tảng mã nguồn mở Gem5.
- Chương 3: Kết quả thực nghiệm và hướng phát triển của đề tài.

**Tóm tắt đề tài: wait for discussing with student.**

Students' info:

- full name: Hoàng Tuấn Anh.
- ID no.: DTC135D5103020003; Điện tử ứng dụng K12A.

>>>>

**#02: Thực thi giải thuật định tuyến mạng trên chip và đánh giá hiệu năng hoạt động của hệ thống mạng trên chip.**

(time: 01/02/2018 to 27/05/2018)

**Mục tiêu:**

- Thực thi mô phỏng một NoC dựa trên nền tảng NOXIM.
- Nghiên cứu và thực thi các thuật toán định tuyến cơ bản trong mạng trên chip, đánh giá hiệu năng hệ thống mạng.
- Làm quen với nền tảng mã nguồn mở trên Linux (Ubuntu) (cái này Students đã pt).

**Nội dung chính:**

- Chương 1: Các giải thuật định tuyến trong mạng trên chip.
- Chương 2: Cơ sở lý thuyết và nền tảng mã nguồn mở NOXIM.
- Chương 3: Kết quả thực nghiệm và hướng phát triển của đề tài.

**Tóm tắt đề tài: wait for discussing with student.**

Students' info:

- full name: Tạ Văn Thư.
- ID no.: DTC135D5103010166; Điện tử ứng dụng K12A.

>>>>

[#Machine Learning et al](#)

>>>

**# useful links:**

!#

some basic definitions and a full picture of AI from past to the future

<https://medium.com/machine-learning-for-humans/why-machine-learning-matters-6164faf1df12>

>>>

**# Đôi điều về học máy (machine learning).**

!#

Phần sau đây là những chủ đề (topics) hay các đoạn (paragraphs), những điều mà tôi thấy cần thiết và hay, có liên quan đến học máy (ML) để hệ thống lại kiến thức bản thân góp nhặt được trong quá trình đọc tài liệu và thực hành thông qua việc coding cũng như dịch thuật (translating).

# Tại sao học máy?

>>>

>>>

## # Game Theory (Lý thuyết trò chơi).

>>>

**#Notes:** một câu chuyện nhỏ về GT!

Lý thuyết trò chơi được ứng dụng ở khắp mọi nơi trong cuộc sống. Ví dụ như việc bạn được lợi ích nhiều từ một sự việc nào đó thì cũng có khả năng là quyền lợi và sự ảnh hưởng đến lợi ích của người khác cũng bị ảnh hưởng. Tuy nhiên, nếu xét về mặt nào đó (cấu hình lại luật chơi) thì có thể cả hai bên sẽ cùng có lợi. Đi cụ thể hơn về một ví dụ như sau: Trong một bộ môn có 2 giáo viên là thầy A và thầy B, cùng phụ trách một môn học C. Vậy việc phân thầy A dạy môn C thì rõ ràng đơn giản là quyền lợi thầy A hưởng và thầy B bị mất quyền lợi. Tuy nhiên, nếu trong trường hợp thầy A ko có nhu cầu dạy, mà thầy B lại có nhu cầu thì rõ ràng việc phân như trên là cả hai người đều bị coi là thua cuộc. Chỉ duy nhất khi thầy B ko có nhu cầu, thầy A có nhu cầu dạy, khi đó thì cả hai người mới được coi là chiến thắng. Do vậy, trong thực tế, có thể tạm phân làm 3 loại game: cùng thắng; cùng thua; người thắng và người thua;

## # Useful links and ref.

>>>

### **#Intro.**

This part is used to store the keys and links that can dev. Ideas and for projects management

>>>

### **# Các giáo sư**

Links dưới đây là thông tin của các giáo sư trong ngành cũng như các giáo sư nổi tiếng

#01:

<http://www.seas.upenn.edu/~andre/>

#02:

>>>

### **# Online courses-resources**

01:EECS 571: Principles of Real-Time Embedded Systems (fall 2010)

Link: <http://www.eecs.umich.edu/courses/eecs571/lectures.html>

02: EE281 - Embedded System Design Laboratory

link: <https://web.stanford.edu/class/ee281/lectures.html>

03: FPGA Design Flow using Vivado

<https://www.xilinx.com/support/university/vivado/vivado-workshops/Vivado-fpga-design-flow.html>

>>>

### **#StackExchange site**

Link:

<https://stackexchange.com/>

username: vnd\_es

email: [namdv2586@gmail.com](mailto:namd2586@gmail.com)

password: normal\_newnumber

>>>

>>>

### **#Keys: Fault-Tolerance NoC (detection, correction, etc).**

Link1: related to SocDep2...

<https://github.com/siavooshpayandehazad/SoCDep2/wiki/CT-Architecture-Exploration>

Link 2: Related to Fault classification with SoCDep2.

<https://github.com/siavooshpayandehazad/SoCDep2/wiki/Fault-Classification>

Link 3: related to bonfire fault classification

<https://github.com/Project-Bonfire/Bonfire/wiki/Fault-Classification>

Link 4: Bonfire fault detection

<https://github.com/Project-Bonfire/Bonfire/wiki/Fault-Detection>

Link 5: Bonfire wiki homepage

<https://github.com/Project-Bonfire/Bonfire/wiki>

>>>

### **#List of links and ref. for top of researching**

#### **#Key: cần đọc những bài báo top cao list sau:**

[https://scholar.google.com/citations?](https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng_computerhardwaredesign)

[view\\_op=top\\_venues&hl=en&vq=eng\\_computerhardwaredesign](https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng_computerhardwaredesign)

Link khác: for conference!

[https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/List\\_of\\_computer\\_science\\_conferences.html](https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/List_of_computer_science_conferences.html)

>>>

#Links ranking Conferences Tops! Updating on 12<sup>th</sup> February 2018.

<http://www.guide2research.com/special-issues/>

<http://www.guide2research.com/journals/>



<http://www.guide2research.com/journals/hardware-electronics>  
<http://www.guide2research.com/topconf/hardware-electronics>  
<http://www.guide2research.com/topconf/>

>>>

>>>

### **# Template of libreoffice impress**

>>>

we can download from this link.ref4

<https://dohliam.github.io/libreoffice-impress-templates/>

>>>

>>>

>>>

### **#How to research?**

Just think and read some similar link:

<https://www.wikihow.com/Get-Started-With-a-Research-Project>

>>>

### **#IPhone account**

Email: [thieuhiep2003@gmail.com](mailto:thieuhiep2003@gmail.com)

Pass: dinhvanha

Username: [thieuhiep2003@gmail.com](mailto:thieuhiep2003@gmail.com)

Pass: dinhvannam2586A.

>>>

### **#link for compiling any eda languages!**

<https://www.edaplayground.com/>

>>>

### **#Useful link for matrix learning!**

<https://www.mathsisfun.com/algebra/matrix-introduction.html>

>>>

### **#How to become an Embedded System/Software Engineering**

<https://www.linkedin.com/pulse/5-differences-between-embedded-maharajan>

there are some useful links in this site you can read more!

>>>

### **#Ref. For Ubuntu 16.04**

[https://www.server-world.info/en/note?os=Ubuntu\\_16.04&p=samba&f=3](https://www.server-world.info/en/note?os=Ubuntu_16.04&p=samba&f=3)

>>> should re-read

### **#How to research effectively**

read the following link

link 1: instructables.

<http://www.instructables.com/id/Researching-a-Research-Paper-Quickly-and-Effective/>

>>>

### **#Links for downloading books and ref. Books**

link 1: libgen

<http://libgen.io/>

link 2: bookz.

<http://bookzz.org/>

updated to

<http://b-ok.org/>

link 3: book4you.

<http://book4you.org/>

email: [namdv2586@gmail.com](mailto:namdv2586@gmail.com)

pass: normal. (should be checked again later!)

### **#keys: RTOS, a comparison of RTOS ...**

!#

Link 1: đọc để hiểu qua khái niệm RTOS.

<https://voer.edu.vn/m/he-dieu-hanh-cho-cac-he-thong-nhung-hdh-thoi-gian-thuc/07006533>

Link 2:

>>>

### **#Links for Projects development.**

>>>

### **#Electronics projects**

link 1:

<http://electronicsforu.com/category/electronics-projects/hardware-diy>

link 2:

<http://www.electronicshub.org/iot-project-ideas/#>

### **#LabVIEW:**

Link 1:

<http://electronicsforu.com/?s=labview>

### **#ARM**

link 1:

<http://www.electronicshub.org/arm-based-projects/>

#>>>

#distraction for today (4R techniques)

### **#keys: pomodoro;**

How to speak E in 2 months

<https://www.quora.com/How-can-I-start-speaking-English-like-a-native-speaker-or-near-in-two-months>

>>>

### **#PICVN**

link:

<http://www.picvietnam.com/forum/showthread.php?t=828>

username: picvn\_namdv

password: normal

email: [namdv2586@gmail.com](mailto:namdv2586@gmail.com)

>>>

>>>

## FPGA and VLSI

>>>

### **# Basic concepts of VLSI up2date**

The link:

<http://www.vlsi-expert.com/p/recommended-book.html>

# setup time and hold time in timing analyze

<http://vlsi.pro/sta-setup-and-hold-time-analysis/>

nice!

>>>

### **#other link for VLSI front-end, back-end engineering tasks.**

The link: key: SDC = Synopsys Design Constraints; SDF: Standard Delay File.

<http://vlsi.pro/synopsys-design-constraints-sdc/>

>>>

### **#Nice link for nice courses**

#1: FPGA+Vivado+...

<https://www.udemy.com/learn-fpga-xilinx-vivado-design-suite-training-in-under-1-hour-vhdl/?instructorPreviewMode=guest>

>>>

### **#can be used for studying and up2date!**

>>>

link

<http://asic-soc.blogspot.com/p/verilog.html>

>>>

### **#for basic and advanced knowledge and technologies in VLSI I think!**

>>>

link!

<http://www.vlsi-expert.com/>

>>>

### **#nice news and nice things that related to VLSI**

!#

<http://www.vlsifacts.com/>

>>>

### **#a nice topic that should read more (for chip design, processor design, fault tolerant design)**

keys: FPGA processor Opencores

Link!

<http://cpu-ns32k.net/>

reference more from open core

<https://opencores.org/project/m32632>

>>>

more link.ref4: where can download various datasheets of processor NX32xxx families.

<http://cpu-ns32k.net/Documents.html>

>>>

BR!

>>>

### **#Start up with VLSI should know!**

!#

<https://www.vlsisystemdesign.com/inception-content-vsd/>

>>>

### **# Sự ảnh hưởng của cặp điện dung lên hệ thống chip vi mạch.**

!#

<https://www.vlsisystemdesign.com/effect-of-coupling-capacitance/>

>>>

>>>

**#? Opencores.... Ref. (where can get onpen sources for doing nice things)**

!#>>>

link ref. <https://opencores.org/>

**username:** vnd171117.

link via account

Username	vnd171117
Fullname	,
Email	dvnd171117@gmail.com
City	HaNoi
State	
Country	
Account created	Mar 17, 2018
Last logged in	Mar 19, 2018

**password:** normal\_newnum.

>>>

**# Xilinx account!**

!#

Username: namdv

email: [namdv2585@gmail.com](mailto:namd171117@gmail.com)

password: [normal@123](#)

link. Ref. <https://www.xilinx.com/myprofile.html>

>>>

**#ARTIX-7 FPGAAC701**

link.ref4

<https://www.xilinx.com/products/boards-and-kits/ek-a7-ac701-g.html#documentation>

link2:

<https://www.xilinx.com/support/documentation-navigation/design-hubs/dh0030-ac701-evaluation-kit-hub.html>

note: link2 get from hhh!

>>>

**#the other link for vlsi + fpga + intel + quatus**

>>>

Link1:

<https://www.altera.com/products/design-software/fpga-design/quartus-prime/support.html>

>>>

link2:

learn more about FPGA

[https://www.altera.com/products/fpga/new-to-fpgas/resource-center/overview.html?](https://www.altera.com/products/fpga/new-to-fpgas/resource-center/overview.html?utm_source=Altera&utm_medium=link&utm_campaign=Homepage&utm_content=Staircase)

[utm\\_source=Altera&utm\\_medium=link&utm\\_campaign=Homepage&utm\\_content=Staircase](https://www.altera.com/products/fpga/new-to-fpgas/resource-center/overview.html?utm_source=Altera&utm_medium=link&utm_campaign=Homepage&utm_content=Staircase)

>>>

or short link

<https://bit.ly/2NTsuth>

>>>

>>>

one more link for generating hdl code from matlab

<https://www.mathworks.com/products/hdl-coder.html>

other link.

<https://www.mathworks.com/help/hdlcoder/examples/getting-started-with-matlab-to-hdl-workflow.html>

!nice!

username: [vndprogrammings@gmail.com](mailto:vndprogrammings@gmail.com)

pass: Dila\*\_newnum

>>>

**# LCD control FPGA ref.**

link.ref4

<https://g3nius.org/lcd-controller/>

>>>

nice!

>>>

**# CNN on FPGA (convolutional Neural Network)**

>>>

link reference

<https://github.com/dgschwend/zynqnet>

>>>

## #ARM

>>>

nice pic.

### Cortex-M Series Family

Smallest and lowest power Cortex processors - optimized for deterministic real time embedded processing and microcontroller applications

The ARM® Cortex®-M processor family is a range of scalable, energy efficient and easy to use processors that meet the needs of tomorrow's smart and connected embedded applications. The processors are supported by the world's #1 embedded ecosystem and have already been shipped in tens of billions of devices. Cortex-M processors help developers deliver more features, in less time, at a lower cost, with versatile connectivity, comprehensive code reuse, standard security and state of the art energy efficiency.

ARM offers:

- Cortex-M0, Cortex M0+, and Cortex-M23 for applications requiring minimal cost, power and area
- Cortex-M3, Cortex-M4, and Cortex-M33 for all applications where a balance between 32-bit performance and energy efficiency is desirable
- Cortex-M7 is designed for embedded applications requiring high performance

The Cortex-M23 and Cortex-M33 include ARM TrustZone® technology for solutions that require optimized, efficient security.

The ARM Cortex-M4, Cortex-M7 and Cortex-M33 processors integrate Digital Signal Processing (DSP) and floating point operations for fast and power-efficient algorithm processing of digital signal control applications such as sensor fusion, motor control and power management.

### #Link for updating all necessary.

<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0646a/index.html>

### #ARM tool comparison (should be updated and reviewed) (Keil MDK-ARM and ARM DS-5)

<http://www.keil.com/arm/rvcomparison.asp>

### #ARM Developer (DS-5)

<https://developer.arm.com/products/software-development-tools/ds-5-development-studio/resources/get-started>

### #Cortex-M into FPGA

link1:

<https://www.community.arm.com/processors/b/blog/posts/prototyping-of-arm-cortex-m-processor-systems-in-fpga>

link2:

<https://www.arm.com/products/designstart/index.php>

link3:

<https://www.arm.com/products/processors/cortex-m/cortex-m1.php>

>>>

# nice community of ARM

>>>

username: [vndprogrammings@gmail.com](mailto:vndprogrammings@gmail.com)

password: vndES\_newnum

web add:

<https://developer.arm.com/>

nice! Should be up2date.!

>>>

>>>

## # Electronic and Circuits

>>>

### **#Một số các câu hỏi về điện tử cơ bản!**

**#01:**

Tại sao trong mạch khuếch đại thuật toán lý tưởng thường có trở kháng vào rất lớn và trở kháng ra rất nhỏ? (tham khảo từ trang 650 sách Floyd (p668of976)).

**#02:**

Khái niệm phối hợp trở kháng được hiểu như thế nào cho đủ và đúng?

**#03:**

Như ta đã biết, trong bán dẫn thuần, dòng lỗ trống tồn tại trong dải hóa trị, trong khi dòng điện tử tự do trong vật liệu bán dẫn thì tồn tại trong dải dẫn. Như vậy, với các loại bán dẫn pha tạp (loại P và N), thì dòng đa số là các lỗ trống và dòng điện tử cũng thuộc hai dải năng lượng tương ứng là dải hóa trị và dải dẫn?

>>>

### **#Các từ khóa trong Proteus**

Cuộn dây: conductor / inductor.

Biến áp: tran/transformer.

>>>

### **# A nice link for electronics**

<http://www.learnabout-electronics.org/index.php>

>>>

### **#Link về mạch điện tử (mô phỏng)**

<https://circuits.io/>

email đk: [dvu.dvt.tn@gmail.com](mailto:dvu.dvt.tn@gmail.com)

pass: diladenxxx\_with\_number

>>>>

### **#Link to get new doc. For circuits.**

!#

<http://talkingelectronics.com/projects/50%20-%20555%20Circuits/50%20-%20555%20Circuits.html>

>>>

#Web links về các khái niệm và kiến thức cơ bản liên quan đến điện tử.

>>>

<https://sites.google.com/site/t2vietdtk/Downhome/gi/8051/vdk/bo-dem-bo-dinh-thoi-trong-8051>

Các kiến thức cơ bản về vi điều khiển, vi xử lý, các linh kiện bán dẫn cơ bản (diode, transistors)...

>>>

## # Notes and general ideas

### # What skills are needed for an ES engineer.

(focus on the Firmware programming)

!#

Should follow this link for dev.

<https://www.quora.com/What-are-skills-needed-to-be-a-successful-firmware-engineer>

### #Ý Tưởng kinh doanh

!01: Link tham khảo về cách kiếm tiền online!

<http://affiliatemarketing.vn/>

>>>

!02: tìm hiểu về affiliate

<https://kiemtiencenter.com/chon-private-program-hay-public-network/>

>>>

### #some links for Hamming codes and related to it!

#link 1:

<http://www.computing.dcu.ie/~humphrys/Notes/Networks/data.error.html#limit>

#Link 2:

<http://www.computing.dcu.ie/~humphrys/Notes/Networks/data.hamming.html>

#Link 3:

### #Notes for Hamming codes

#Note 1:

#### VHDL Code of (7,4) Hamming Code Encoder

 Share 4  Like 4  Tweet  Pin it  G+  Share

Name of Pins	Direction	Width
Datain	Input	4
Hamout	Output	7

Hamming code is useful in Error Correction in Linear Block Code. This code will encode four bits of data and generate seven bits of code by adding three bits as parity bits. It was introduced by Richard W. Hamming. This algorithm can detect one and two bit error and can correct one bit error. Given below code will generate (7,4) Systematic Hamming Encoder. This encoder will use Least Significant 4 bits as data inputs and Most 3 significant bits as a parity bits.

Parity bits equations are given below

$p_0 = \text{datain}(0) \text{ xor } \text{datain}(1) \text{ xor } \text{datain}(3)$

$p_1 = \text{datain}(0) \text{ xor } \text{datain}(2) \text{ xor } \text{datain}(3)$

$p_2 = \text{datain}(1) \text{ xor } \text{datain}(2) \text{ xor } \text{datain}(3)$

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
```

ref. From. Link: <https://vlsicoding.blogspot.com/2015/12/vhdl-code-of-74-hamming-code-encoder.html>

#Note 2:



>>>

>>>

### **# Cách lưu file ảnh định dạng vector**

>>>

Bước 1: Vẽ hình dưới dạng bảng biểu hay bất kể hình gì đó bằng tools (libre office)...

Bước 2: Save as hay export trực tiếp ra file file.pdf

Bước 3: Sử dụng chức năng crop của công cụ master pdf: mở file >> document >> crop pages...

BR!

>>>

## # SISLAB Infor.

#>>>

### #PHD information Notes

#Link courses on web

<https://courses.uet.vnu.edu.vn/>

username: 17028023

password: normal.

#Link email:

>>>

### # Báo cáo tiến độ công việc (tuần, tháng, quý, năm)

>>>

### # Báo cáo công việc tuần 3 (13/8 17/8/2018)

>>>

Báo cáo tiến độ công việc (Tuần 3: 13/8 đến 17/8)

1) Công việc hoàn thành:

1.1. Hoàn thành báo cáo các khái niệm cơ bản trong hệ thống có khả năng chống lỗi .

1.2. Phân loại các kỹ thuật phát hiện, sửa lỗi trong mạng trên chip (Time/Information/Spatial Redundancy techniques).

2) Công việc đang thực hiện:

2.1. Các kỹ thuật cơ bản sử dụng trong hệ thống truyền thông trên chip có khả năng chống lỗi.

2.3. Review bộ mô phỏng NoC sử dụng Garnet2.0.

3) Công việc dự kiến trong tuần tiếp theo.

3.1. Review công cụ tổng hợp (DC).

3.2. Nghiên cứu về mô hình lỗi sử dụng trong mạng Garnet2.0.

cập nhật trên redmine vào ngày 18/8/2018

>>>

### #How to get the grade.

>>>

<http://112.137.129.30/viewgrade/>

>>>

>>>

!#

Danh sách email gửi tham khảo giá

<http://suno.vn/blog/top-10-dia-chi-cung-cap-van-phong-pham-gia-goc-o-ha-noi-va-tphcm/>

>>>

up-to-date.

### #some important information (coffee, papers, sugar... etc)

# số điện thoại mua coffee (liên hệ a Phong; a. Vũ).

# Số điện thoại thay mực máy in

# Số điện thoại gọi nước uống Lavie

# Số điện thoại etc

>>>

### #SISLAB IEEE membership

username: van-nam dinh

link:

<https://ieeexplore.ieee.org/Xplore/home.jsp>

username: Van-Nam Dinh

password: sislab2018

→ normarpass\_newnum.

email: [dvnam@ictu.edu.vn](mailto:dvnam@ictu.edu.vn)

>>>

### #add. for weekly report!

<http://10.10.1.61/redmine/issues/243>

user: namdv  
password: normal\_with\_number.  
**#add. Wiki PTN.**  
[http://10.10.128.33/wiki/index.php/Main\\_Page](http://10.10.128.33/wiki/index.php/Main_Page)  
user name and password are the same the above.  
<http://10.10.128.33/wiki/index.php/User:Namdv>  
>>>

### **# 1 way to access to user-share**

>>>  
link:  
<smb://10.10.128.33/user-share/>  
username: namdv  
pass: normal.  
>>>

### **#Copy and paste between hosts**

use the following commands

a) from host to your computer.

```
scp namdv@10.10.128.41:/home/userdata/hvch/namdv/Makefile ../
```

and b) from your computer to host.

```
scp -r Ann/ namdv@10.10.128.41:/home/userdata/hvch/namdv/
```

or:

```
scp -r *.vhd namdv@10.10.1.64:/home/userdata/hvch/namdv/hdl-codes/
```

they are work and updating on today is Thursday 8<sup>th</sup> February 2018

>>>

### **#SSH in Lab**

one exapple

```
ssh -Y namdv@10.10.1.64
```

other adds: 10.10.128.41 or 10.10.1.61 or 10.10.128.33

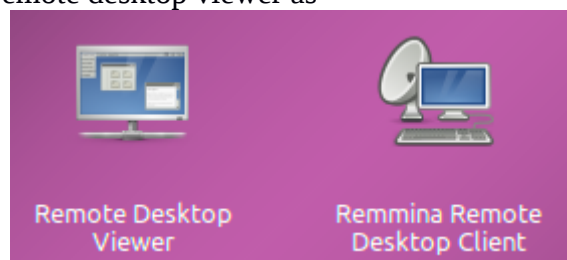
>>>

### **#more interesting**

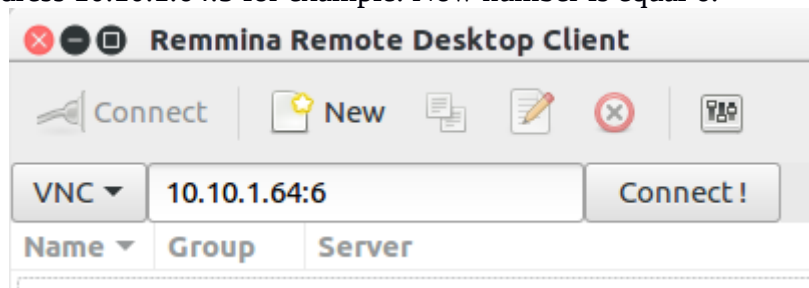
```
vncserver -geometry 1920x1200
```

after doing this command will get the number like 3, 4, 5... for instance.

Use tool from ubuntu like remote desktop viewer as



then chose vnc address 10.10.1.64:5 for example. Now number is equal 6.



current using in real life! 6=>5

#after running it, turn-off it by

```
vncserver -kill :5
```

Enjoy it!

>>>

### #How to accept into share folder at SISLAB?!

!# look at the following picture.

>>>

### #How to use ModelSim on server?

Step1: go inside your account via ssh

```
ssh -Y namdv@10.10.128.41
```

pass:

Step2: go inside tool via commands

```
cd /home/apps/mentor/Questasim/10.1d/questasim/linux_x86_64/
```

Step3: source your env.sh file to build your environment to run.

```
source /home/apps/mentor/env.sh
```

Step4: type vsim

:( step4 is not working! :(

chú ý: không chạy là do ban đầu trong bước 1 là -y, chạy khi sửa thành -Y.

>>>

### #synopsys Training course HOW2USE?

Use ssh command to go inside adds: .33, .41 and look for /home/apps/synopsys/training/  
ref. from. HieuBD

```
Cấu trúc thư mục:
/home/apps/synopsys/training
├── DCCompiler
│   ├── DC1_2009.06
│   ├── DC1_200906_JobAid.pdf
│   ├── DC1_200906_LG.pdf
│   └── DC1_200906_SG.pdf
├── ICCompiler
│   ├── ICC1_200906_Job_Aid.pdf
│   ├── ICC1_200906_LG.pdf
│   ├── ICC1_200906_SG.pdf
│   └── IC_Compiler_2009.06
├── PrimeTime
│   ├── 01_PT_intro.pdf
│   ├── 02_PT_Day1.pdf
│   ├── 03_PT_Day2part1.pdf
│   ├── 04_PT_Day2part2_Day3.pdf
│   ├── Lab1_PT1_2009.12_LG.pdf
│   ├── Lab2_PTSI_200612_LG.pdf
│   ├── Lab3_PrimeTime_PX_200906_LG.pdf
│   ├── PT_1_2009.12
│   ├── PT_PX_2009.06
│   └── PT_SI_2007.06
└── vcsmx
    ├── VCSMX_training.pdf
    └── vcsmx_lab_2008
```

>>>

### #How to run example? (CD compiler of synopsys)

After get this folder, we can follow these steps to run.

```

2 http://10.10.1.61/repos/Giants/branches/DuyHieu/freepdk45_ref_flow/
3
4 http://10.10.1.61/repos/Giants/branches/DuyHieu/freepdk45_ref_flow/
5
6 git svn clone http://10.10.1.61/repos/Giants/branches/DuyHieu/freepdk45_ref_flow/

```

- ✓ Go to that folder via command `cd`; for examples path =  
/home/userdata/hvch/namdv/WORK/freepdk45\_ref\_flow
- ✓ source file `env.sh` via commands `source env.sh`
- ✓ go to the **syn** folder. Then run **make** command.
- ✓ Look for netlist file outputs in the **results** folder. (vim file \*.v to see your design).

For more detail:

[http://10.10.1.61/repos/Giants/branches/lsi\\_contest\\_2018/](http://10.10.1.61/repos/Giants/branches/lsi_contest_2018/)

it should have an account to access!

!nice!

>>>

**#repository where I am working with (updating on 4 May 2018)**

>>>

<http://10.10.1.61/repos/sistj/trunk/impl/syn/>

>>>

`git svn add...`

`cd syn`

Example: (to add 2 files)

`git add 2files...`

`git commit`

`git svn dcommit`

>>>

**#how to update in this...**

>>>

`git svn rebase`

`git svn dcommit`

>>>

If the above commands have problems, can try with  
`git tash...`

**#How to run a DC for your own design.**

>>>

**#announcement from Prof. Tu TX**

[Feature #292: NEWCOMMERS: Guidelines for newcomers at SISLAB](#)

- Author: Xuan-Tu Tran
- Status: New
- Priority: Normal
- Assignee: Xuan-Tu Tran
- Category:
- Target version:

Đây là một số thông tin cơ bản và yêu cầu khi bắt đầu làm việc tại PTN SIS ạ.

1. Đọc Wiki của PTN tại địa chỉ: [10.10.128.33/wiki](http://10.10.128.33/wiki). Địa chỉ này chỉ có thể truy cập bằng máy tính của PTN hoặc sử dụng IP đã được cấp. Tài khoản chung cho sinh viên đối với các máy để bàn tại PTN: Account: student Pass: Student
2. Quy định làm việc tại PTN SIS: [http://10.10.128.33/wiki/index.php/Quy\\_%C4%91%E1%BB%8Bnh\\_l%C3%A0m\\_vi%E1%BB%87c\\_t%E1%BA%A1i\\_SIS\\_Lab](http://10.10.128.33/wiki/index.php/Quy_%C4%91%E1%BB%8Bnh_l%C3%A0m_vi%E1%BB%87c_t%E1%BA%A1i_SIS_Lab)
3. Cài ubuntu và lập trình C/C++ trên ubuntu. file guide-linux.txt
4. Sách C/C++: "The C Programming language"

**Cách truy cập tài liệu phòng TN SIS**

**# Lệnh điều khiển máy ảo của Lab**

`rdesktop -f -N -u namdv -d sislab 10.10.128.25`

>>>

để có thể thực hiện chuyển đổi giữa giao diện màn hình máy cá nhân khi đang dùng máy ảo, sử dụng tổ hợp phím `trl+alt+enter`.

***can use the following commands***

```
rdesktop-vrdp -u namdv -d sislalab 10.10.128.25
```

or

```
rdesktop -u namdv -d sislalab 10.10.128.25
```

or

```
rdesktop -u namdv -g 2400x1200 -d sislalab 10.10.128.25
```

>>>

**one more way to use remote desktop**

```
xfreerdp -f -u namdv -d sislalab 10.10.128.63
```

>>>

or

```
xfreerdp -u <SIS_username> -d sislalab -w 2048 -g 1400 10.10.128.63
```

or

```
xfreerdp -f -u namdv -d sislalab -g 2048x1400 10.10.128.63
```

or

```
xfreerdp -u namdv -d sislalab -g 2048x1400 10.10.128.63
```

>>>

for more detail and help

```

See http://www.rdesktop.org/ for more information.

Usage: rdesktop [options] server[:port]
-u: user name
-d: domain
-s: shell / seamless application to start remotly
-c: working directory
-p: password (- to prompt)
-n: client hostname
-k: keyboard layout on server (en-us, de, sv, etc.)
-g: desktop geometry (WxH)
-i: enables smartcard authentication, password is used as pin
-f: full-screen mode
-b: force bitmap updates
-L: local codepage
-A: path to SeamlessRDP shell, this enables SeamlessRDP mode
-B: use BackingStore of X-server (if available)
-e: disable encryption (French TS)
-E: disable encryption from client to server
-m: do not send motion events
-C: use private colour map
-D: hide window manager decorations
-K: keep window manager key bindings
-S: caption button size (single application mode)
-T: window title
-t: disable use of remote ctrl
-N: enable numlock synchronization
-X: embed into another window with a given id.
-a: connection colour depth
-z: enable rdp compression
-x: RDP5 experience (m[odem 28.8], b[roadband], l[an] or hex nr.)
-P: use persistent bitmap caching
-r: enable specified device redirection (this flag can be repeated)

```

should be use other option

```
rdesktop -f -u namdv -d sislab 10.10.128.25
```

```
>>>>
```

### #other notes:

Fwd: [Sislab Admin] Account information for sislab computers and resources

Inbox x



Quang Nguyen Kham Hong

14:32 (1 hour ago) ✉

to me ▾

----- Forwarded message -----

From: Administrator <[scamemiv@gmail.com](mailto:scamemiv@gmail.com)>

Date: 2017-11-21 9:34 GMT+07:00

Subject: [Sislab Admin] Account information for sislab computers and resources

To: [khamquang@gmail.com](mailto:khamquang@gmail.com)

Dear Nguyen-Kham-Hong Quang,

This is your account information for accessing SIS Lab computers and resources:

Username: quangnh

Password: W2o0kkeU4N0=

Your storage quota: 4G

You should change the password immediately by using this url: <http://10.10.1.61/chpass/chpasswd.cgi/index/> (remember that this url is only accessed in the UET network)

Accessing file server: \\10.10.128.33 (windows) or smb://10.10.128.33 (linux and Unix)

Project Management Page: <http://10.10.1.61/redmine>

Some materials are at: <http://10.10.128.33/wiki>

```
>>>>
```

**Passwifi SISLAB: n0p4sswd**

```
>>>>
```

### #How to use printer at SISLab?

```
!#
```

```
sudo vim /etc/cups/client.conf
```

```
ServerName 10.10.128.33
```

enjoy it!

>>>

**#H3**

#site

<https://yoloh3.github.io/>

#nice

**#PHD's Topics and targets (via keys)**

**#01: Gem5**

The gem5 simulator is a modular platform for computer-system architecture research, encompassing system-level architecture as well as processor microarchitecture

Links ref:

[http://www.gem5.org/Main\\_Page](http://www.gem5.org/Main_Page)

other links:

<http://gem5user.blogspot.com/2013/02/take-first-step.html>

and

<http://gem5user.blogspot.com/>

youtube link

<https://www.youtube.com/watch?v=fD3hhNnFL6k>

For reading more

<http://learning.gem5.org/book/>

For more and more! (from installing)

<http://learning.gem5.org/book/index.html>.

>>>



## # BSG PROJECT

>>>

### #Link SC của dự án Toshiba

<http://10.10.1.61/repos/sistj/>

dùng lệnh git svn clone ... để tải những thứ cần về.

>>>

### #How to run simulation

S1: ssh to the server with

ssh -Y namdv@10.10.1.64

S2: run env.sh file from path = /home/LOCAL/namdv/bsg\_manycore

source env.sh

S3: run simulation commands from path =

/home/LOCAL/namdv/bsg\_manycore/software/spmd/hello

./simv -gui

or

dve -vpd \*.vpd

Note: you can run make before step3

>>>

# more about many\_cores projects

>>>

link ref. <http://bjump.org/manycore/>

>>>

### #Where to start and up2date?

>>>

## Working progress!

>>>

Tuesday, August 21<sup>st</sup> 2018

>>>

#Task1: Review synthesis tools form synopsys DC's labs guide

>>>

### bug 1: wrong version

as I try follow README file.

```
namdv@hercules: /home/LOCAL/namdv/DC_2015.06$ source setup.sh
namdv@hercules: /home/LOCAL/namdv/DC_2015.06$ ./testscript

*****
Running the Synopsys Workshop Installation Test Script

      Course Name:      Design Compiler
      Course Version: 2015.06
      Last Update:      2016-08-31
*****

Running Test 1 - installation integrity check
Running Test 2 - testing executable(s)
      2a running dc_shell (version needed: 2015.06-SP5)
Test 2a FAILED: wrong version: dc_shell
Test 2b FAILED: Cannot locate fm_shell executable in current search path
Skipping Test 3 - GUI

Installation Test FAILS for Design Compiler workshop
For details execute: cat .Test/results
```

>>>

SOLUTION????

>>>>

## # Task2: Doing LAB1

>>>>

### %Task 1: Examine and modify the setup files

Note 1: what is task1 doing?

ok

### %Task 2: Invoke Design Vision

Q1: What is the Link library?

\* ../ref/libs/mw\_lib/sc/LM/sc\_max.db

Q2: What is the Target library?

../ref/libs/mw\_lib/sc/LM/sc\_max.db

Q3: What is the Symbol library?

../ref/libs/mw\_lib/sc/LM/sc\_sdb

Q4: What user directories have been added to the Search path?

. /home/tools/synopsys/syn/K-2015.06/libraries/syn /home/tools/synopsys/syn/K-

2015.06/minpower/syn /home/tools/synopsys/syn/K-2015.06/dw/syn\_ver

/home/tools/synopsys/syn/K-2015.06/dw/sim\_ver ../ref/libs/mw\_lib/sc/LM/ ./rtl/ ./scripts/

### final step (9)

```
Sanity check for TLU+ vs MW-Tech files:
```

```
max_tlu+: ../ref/libs/tlup/cb13_6m_max.tluplus
```

```
min_tlu+: **NONE**
```

```
mapping_file: ../ref/libs/tlup/cb13_6m.map
```

```
max_emul_tlu+: **NONE**
```

```
min_emul_tlu+: **NONE**
```

```
MW design lib: TOP_LIB_VND
```

```
----- Sanity Check on TLUPlus Files -----
```

```
1. Checking the conducting layer names in ITF and mapping file ...
```

```
[ Passed! ]
```

```
2. Checking the via layer names in ITF and mapping file ...
```

```
[ Passed! ]
```

```
3. Checking the consistency of Min Width and Min Spacing between MW-tech and ITF ...
```

```
[ Passed! ]
```

```
----- Check Ends -----
```

```
1
```

```
design_vision-topo>
```

>>>>

%Task3: Read the design into DC Memory

OK

%Task 4: Explore the Schematic View

prob: don't understand it!

%Task 5: Explore the Mouse Functions

This task helps to know how to explore the design.

%Task 6: Constrain TOP with a Script file

source TOP.con from scripts folder

(source scripts/TOP.con)

This step will execute a script file, which constrains the TOP design. Should I know it?

%Task 7: Compile or Map to Vendor-Specific Gates

This step will help to generate optimal timing and area results. (auto ungrouping)

%Task 8: Generate Reports and Analyze Timing

rc: is an alias that was specified in the .synopsys\_dc.setup file ~ report\_constraint -all\_violators

```

rt: report_timing => shows the timing of the critical path
Slack (Violated): -0.06
ra: report area
Total cell area: 236.25
%Task 9: Save the Optimized Design
bug: write_icc2_files -output ./mapped/TOP_icc2
??
%Task 10: Remove Designs and Exit Design Vision
nothing
%Task 11: nothing
%Task 12: OK
>>>

```

**Wednesday, August 22<sup>nd</sup> 2018**

>>>

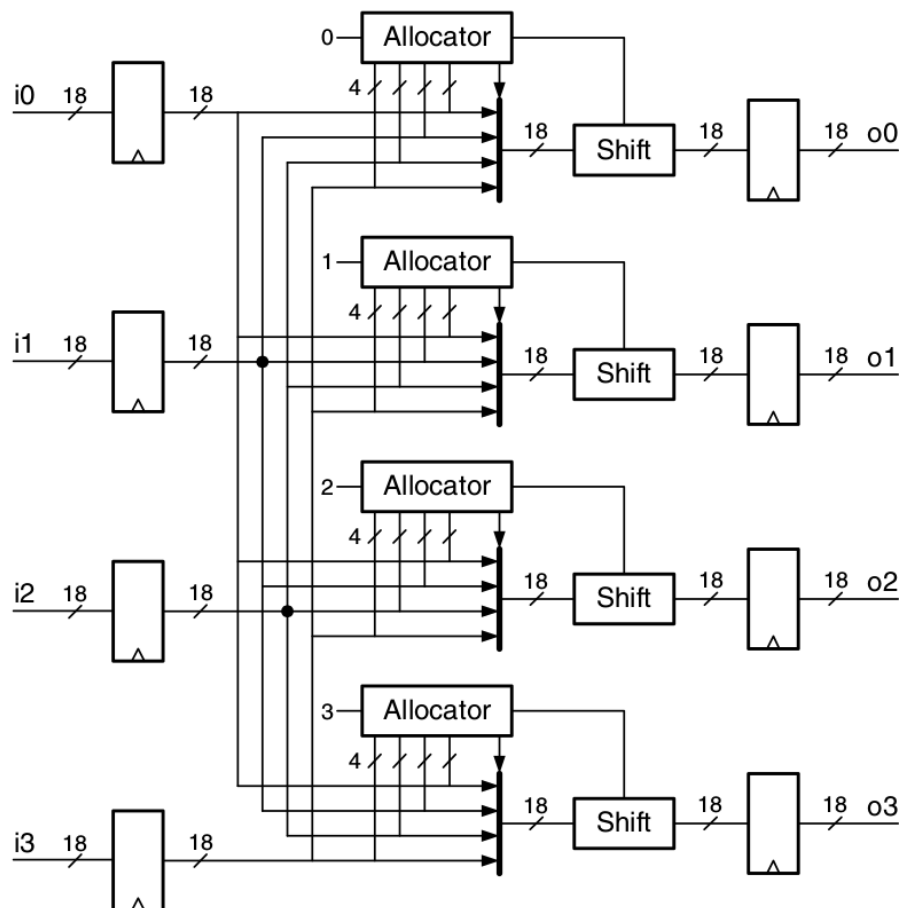
### #TASK1: NoC router building

>>>

%step1: translate into Vnese to understand more!

#### Part 2.5: Thiết kế router

Mỗi một nút chuyển mạch trong mạng butterfly là một router, nó có khả năng nhận các gói tin tại đầu vào, xác định đích đến của nó dựa theo giải thuật định tuyến, sau đó chuyển tiếp gói tin đến lối ra phù hợp. sơ đồ khối của một router đơn lẻ được minh họa như hình dưới đây:

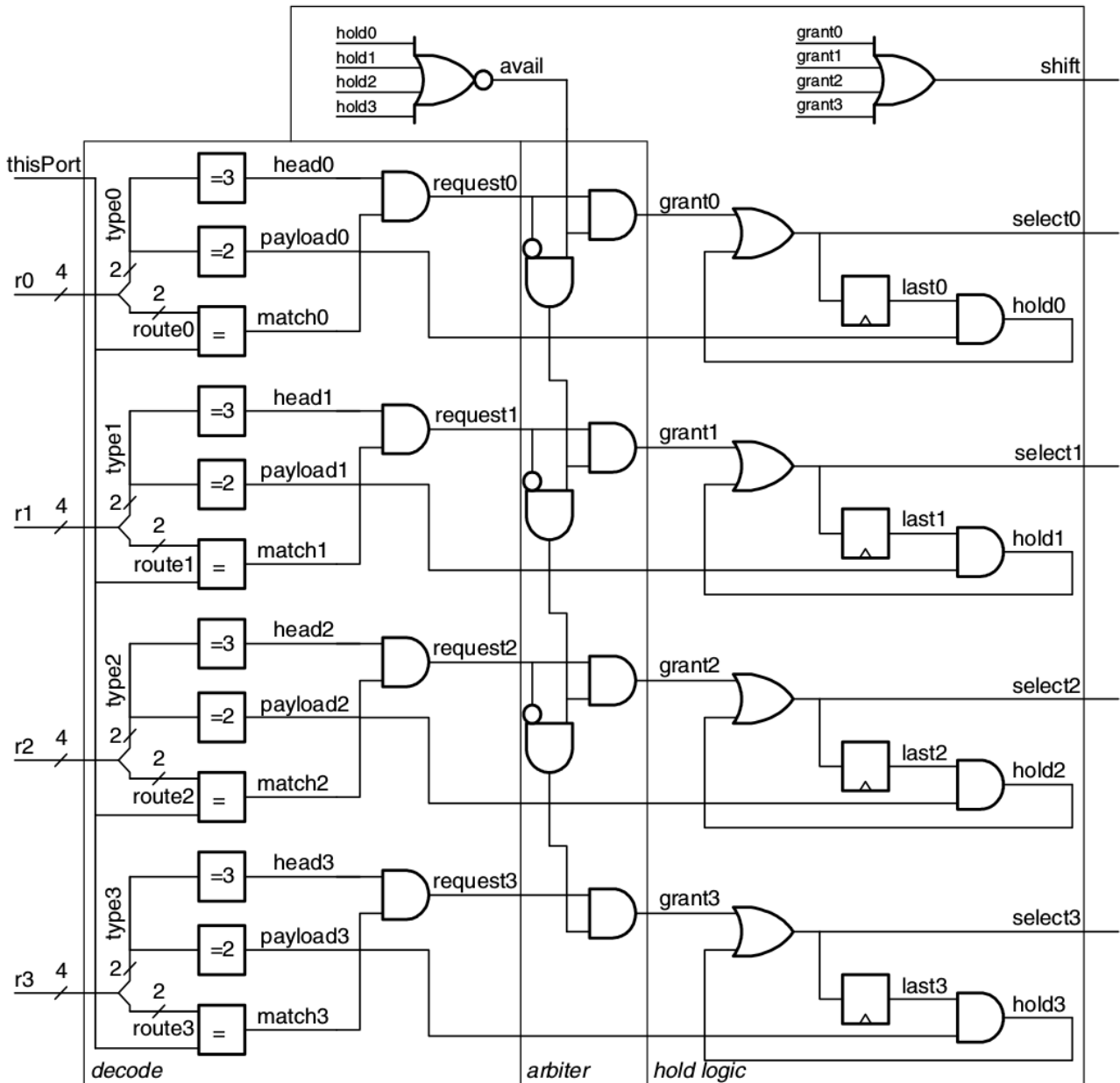


datapath của router gồm bốn thanh ghi lối vào 18 bit, 4 bộ hợp kênh 18 bit (multiplexer) bốn bộ dịch (shifters) (cho việc dịch chuyển đường đi của các header phits), và 4 thanh ghi lối ra 18 bit. Datapath gồm 144 bit thanh ghi và khoảng 650 cổng (2-input NAND-equivalent).

Các phits đến tại mỗi xung clock đưa vào đầu vào các thanh ghi lối vào và được định tuyến tới tất cả bốn bộ multiplexers. Tại mỗi multiplexer, kết hợp với bộ cấp phát (allocator) để xem xét loại phit và hợp tiếp theo của mỗi phit đầu (head phit), thông qua đó sẽ thiết lập chuyển mạch. Các phits từ

đầu vào được lựa chọn sẽ được định tuyến đến các router tiếp theo để dịch chuyển. Dưới sự điều khiển của bộ cấp phát (allocator), bộ dịch (shifter) dịch tất cả các đầu phit (head phits) rồi đi bằng 2 bits để tách rời khỏi trường định tuyến hiện tại, và tìm đến trường định tuyến tiếp theo. Các payload phits được dịch chuyển nguyên vẹn không thay đổi.

Sự điều khiển của các bộ định tuyến nằm hoàn toàn bên trong bốn bộ cấp phát (allocator) kết hợp với mỗi đầu ra điều khiển bộ hợp kênh (multiplexer) và bộ dịch chuyển (shifter). Mỗi bộ cấp phát (allocator), như tên của nó, cấp phát một cổng đầu ra từ một trong bốn cổng đầu vào. Sơ đồ nguyên lý của một bộ cấp phát allocator như hình sau:



Một bộ cấp phát (allocator) bao gồm bốn bit gần giống hệt nhau, được chia thành ba phần: giải mã (decode), phân xử (arbitrate) và giữ (hold). Trong phần giải mã, bốn bit có trọng số cao của mỗi đầu vào phit được giải mã. Mỗi bộ giải mã sẽ cung cấp (phát-generate) 2 tín hiệu. Tín hiệu request thứ i là đúng (true) nếu phit trên đầu vào i là một head phit và hai bit trọng số cao hơn của trường định tuyến (route field) sẽ tương ứng với số của cổng đầu ra. Tín hiệu này xác định rằng đầu vào phit yêu cầu (phit request) sử dụng cổng đầu ra để định tuyến gói tin bắt đầu với head phit. (phit tiêu đề). Bộ giải mã (decoder) cũng phát tín hiệu payload i là đúng (true) nếu phit trên đầu vào i là phit payload. Mức logic giữ (hold logic) của bộ cấp phát (allocator) dùng tín hiệu này để giữ một kênh trong khoảng thời gian của một gói tin. (miễn sao payload phits là được lựa chọn tại cổng đầu vào).

Tầng thứ 2 của bộ allocator là một bộ phân xử thiết lập ưu tiên bốn lối vào. việc thiết lập cố định thứ tự ưu tiên tại bốn lối vào (four-input) như vậy là không được sử dụng trong thực tế, bởi vì nó dẫn đến hiện tượng không công bằng (unfair) giữa các router. điều này có thể dẫn đến livelock hoặc bão hòa (starvation) trong mạng. Bộ phân xử (arbiter) chấp thuận 4 tín hiệu yêu cầu và phát (generate) bốn tín hiệu phân phát (grant = ban). Nếu cổng đầu ra sẵn sàng (được xác định bởi tín hiệu *avail*), bộ arbiter ban phát đến cổng để cổng đầu vào đầu tiên tạo một yêu cầu. Việc xác nhận (asserting) tín hiệu ban phát gây ra tín hiệu *select* tương ứng được xác nhận.

>>>>

**Thursday, August 23<sup>rd</sup> 2018**

>>>>

**%Task 1: Review BSG synthesis flow**

>>>>

**#What am I doing?**

Step 1: run synthesis steps via available scripts.

I try to synthesis BSG many core from

/home/LOCAL/namdv/sistj/trunk/impl/syn

by doing the following commands:

source env.sh

make

>>>>

**notes:**

**#Note1:**

The outputs is file “bsg\_manycore\_hetero\_socket.mapped.v” from

path=“/home/LOCAL/namdv/sistj/trunk/impl/syn/bsg\_manycore\_hetero\_socket.results”

**#An explore:**

in Makefile I note that:

dc\_shell -f ./rm\_dc\_scripts/dc.tcl 2>&1 | tee \${DESIGN\_NAME}.log

=> we consider the *dc.tcl* file from *rm\_dc\_scripts* folder.

Trong file dc.tcl có các nội dung gì cần quan tâm?

source -echo -verbose ./rm\_setup/dc\_setup.tcl

# Define the verification setup file for Formality

set\_svf \${RESULTS\_DIR}/\${DCRM\_SVF\_OUTPUT\_FILE}

# Read in the RTL Design

# Read in the RTL source files or read in the elaborated design (.ddc).

puts "RM-Info: Sourcing script file [which \${DCRM\_RTL\_READ\_SCRIPT}]\n"

source -echo -verbose \${DCRM\_RTL\_READ\_SCRIPT}

# read\_ddc \${DCRM\_ELABORATED\_DESIGN\_DDC\_OUTPUT\_FILE}

current\_design \${DESIGN\_NAME}

link

write -hierarchy -format ddc -output \${RESULTS\_DIR}/\${DCRM\_ELABORATED\_DESIGN\_DDC\_OUTPUT\_FILE}

...

...

# compile\_ultra -gate\_clock

compile

optimize\_netlist -area

# Write out Design

write -format verilog -hierarchy -output \${RESULTS\_DIR}/\${DCRM\_FINAL\_VERILOG\_OUTPUT\_FILE}

write -format ddc -hierarchy -output \${RESULTS\_DIR}/\${DCRM\_FINAL\_DDC\_OUTPUT\_FILE}

# Write and close SVF file and make it available for immediate use

set\_svf -off

# Use SAIF file for power analysis

```
report_power -nosplit > ${REPORTS_DIR}/${DCRM_FINAL_POWER_REPORT}
report_power -hier -nosplit > ${REPORTS_DIR}/${DCRM_FINAL_POWER_REPORT_HIERARCHY}
report_clock_gating -nosplit > ${REPORTS_DIR}/${DCRM_FINAL_CLOCK_GATING_REPORT}
# Write out Milkyway Design for Top-Down Flow
write_milkyway -overwrite -output ${DCRM_FINAL_MW_CEL_NAME}
>>>
```

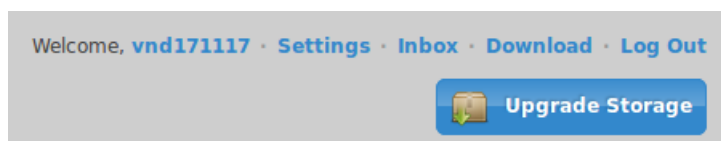
# BOOKs and Reference. (#num.Authors.title.place.year)

>>>

### #Zotero note taking

username: vnd1711117 email: [hocngoangu2586@gmail.com](mailto:hocngoangu2586@gmail.com).

as following figure



pass: normal\_number

link ref:

[https://www.zotero.org/support/kb/keyboard\\_shortcuts](https://www.zotero.org/support/kb/keyboard_shortcuts)

>>>

### #Links for downloading

#link 1:

<http://bookzz.org/>

#link 2:

<http://book4you.org/>

email: [namdv2586@gmail.com](mailto:namdv2586@gmail.com)

pass: normal\_withnum.

#link 3:

<http://b-ok.org/>

#Link 4:

<http://libgen.io/>

#Link 5: download papers.

<http://sci-hub.la/>

or:

<http://sci-hub.tv/>

can check more from

Nghe đồn **SCI HUB** về nhà mới

Em gửi mọi người ạ:

- 1) <http://sci-hub.la/>
- 2) <http://sci-hub.hk/>
- 3) <http://sci-hub.mn/>
- 4) <http://sci-hub.name/>
- 5) [http://sci-hub.tv](http://sci-hub.tv/)
- 6) [http://sci-hub.tw](http://sci-hub.tw/)

>>>

#Link 6:

>>>

#1. Penny Cameron, Vanessa Told; THE NEW Prepare for IELTS – Academic module, University of Technology Sydney, Australia 2001.

>>>

#2. Link for ref. book. (Daniel. Book: Reliable Computer systems - 2<sup>nd</sup> edition (1992)).

<https://www.sciencedirect.com/science/book/9781555580759>

>>>



## # Useful webs

>>>

#short link online

>>>

use link.

<https://bitly.com/>

>>>

**# Một số links thảo luận và nêu quan điểm về vấn đề NCKH ở VN nên lưu lại tham khảo:**  
!#

Link 1:

<http://ired.edu.vn/vn/doc-tin/74/ba-van-de-cua-dai-hoc-viet-nam-hien-nay>

Link 2:

>>>

**# can use it for updating replace to 24h**

<http://www.fullmatchesandshows.com/2018/04/03/juventus-vs-real-madrid-highlights-full-match/>

>>>

other link.ref4

<https://www.reddit.com/r/soccer/>

>>>

**#Đồng bộ trên firefox**

email: [vndprogrammings@gmail.com](mailto:vndprogrammings@gmail.com)

pass: vndprogrammings\_newnum.

>>>

**#typing master (try to practice as many as you can to improve your typing speed)! :)**

link 1:

<https://10fastfingers.com/typing-test/english>

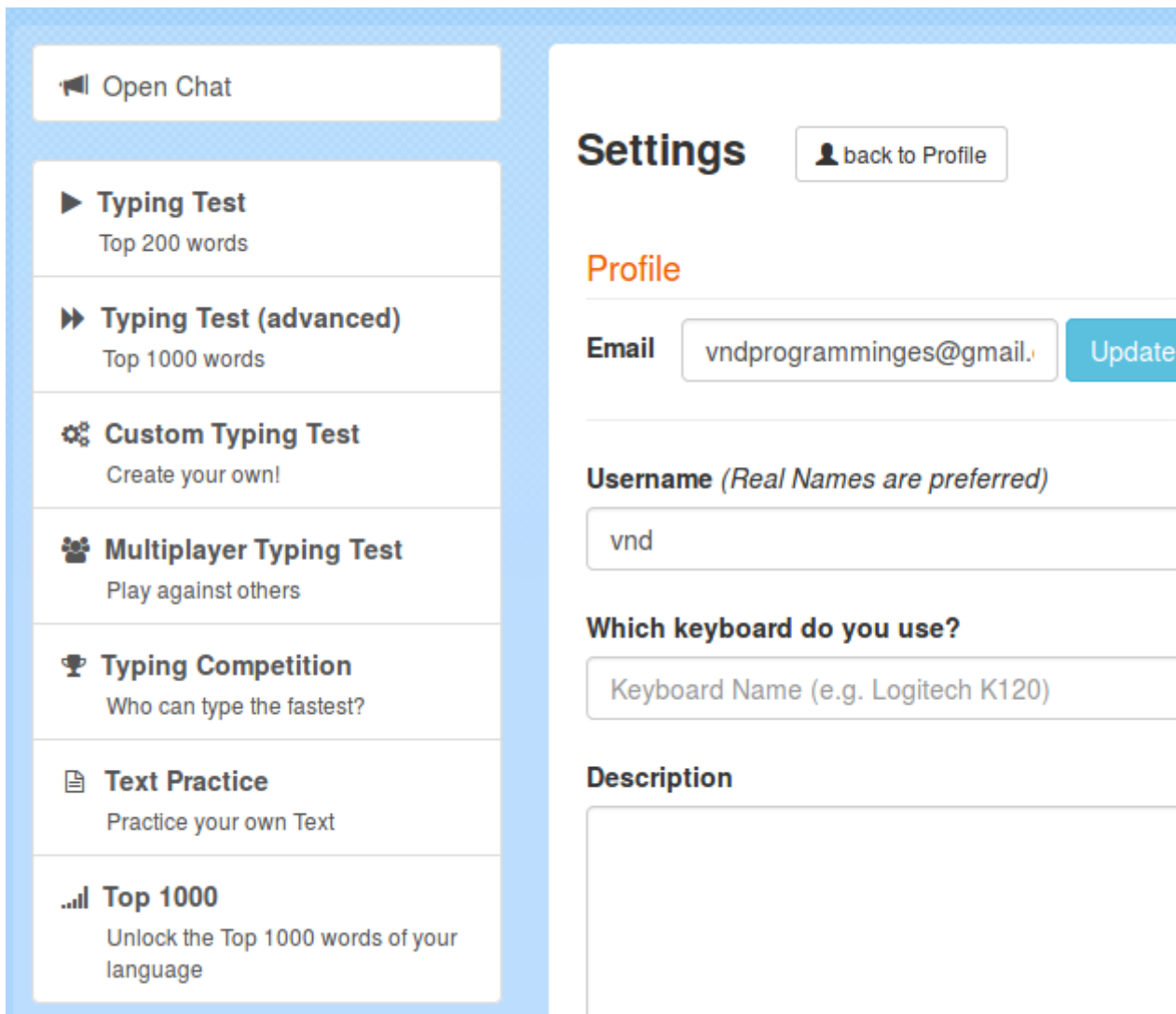
username: vnd.

Email: [vndprogrammings@gmail.com](mailto:vndprogrammings@gmail.com)

password: normal\_oldnum

pic.2.rem

>>>



link 2:

<http://play.typeracer.com/>

username: vnd\_type\_training

password: normal+newnum.

Email: [vndprogrammings@gmail.com](mailto:vndprogrammings@gmail.com).

>>>

**#some useful keys via courses.**

<http://onlinestatbook.com/2/index.html>

I have cared and read about Probability, Research Design.

>>>

**#Q&A**

**#How to remember the book you read**

<https://www.quora.com/How-do-you-remember-the-book-you-read>

It was taken by myself from above link!

I have learned to be strategic about my reading, and manage my memory. I have developed some key reading, learning and memory recall skills. As for any habit change, the key is to form new habits in place of ineffective ones, and that requires initial practice and conscious effort. The good news is that this is easy to implement.

My reading strategy has 3 steps:

**1. Scanning the territory I am exploring, and defining my learning needs.**

I always read with a clear purpose in my mind, which primes the brain to identify and prioritise the key ideas, and I only read books that serve my strategic purpose and which capture my interest and energise me. I don't waste time on poorly written or boring books, I just move on.

**2. Define clear goals before reading every book.**

Set your intention and attention correctly, and you will remember more. Knowing in advance if you are reading to scan for ideas, or to deeply digest content that you know is worth mining for insight, is key. Low productivity is everyone's enemy, eating up our time unnecessarily.

**3. Use methods that work for me to connect, distill and chunk the key ideas in any book.**

**# nice Q&A for my major: keys FPGA and VLSI verification**

<https://www.quora.com/How-can-I-be-a-guru-at-FPGA-VLSI-design-and-verification>

**# Template for writing a CV**

<https://www.cvplaza.com/cv-templates/free-cv-templates/>

>>>

**# Electronics ref.**

# for HW design with altium et al.

<http://www.fedével.com/welldoneblog/>.

>>>

**#Download papers via DOI number:**

<http://sci-hub.cc/>

It should be noted!

- 1) <http://sci-hub.la/>
- 2) <http://sci-hub.hk/>
- 3) <http://sci-hub.mn/>
- 4) <http://sci-hub.name/>
- 5) <http://sci-hub.tv>
- 6) <http://sci-hub.tw>

Base on DOI! :)

>>>

**#How to write a journal article (step-by-step)**

<https://writingajournalarticle.wordpress.com/sections-relevant-to-all-papers/introduction/main>

<https://writingajournalarticle.wordpress.com/>

>>>

**Keys:**

**#VHDL programming et al**

#Links for studying VHDL by tutorial and sample codes

<http://esd.cs.ucr.edu/labs/tutorial/>

>>>

**#Web dientuvietnamdotnet**

!#hiz đăng ký!

<b>Tên truy cập</b>	<input type="text" value="vnd1171117"/>
	<small>Please enter the name by which you would like to log-in and be known on this site.</small>
<b>Mật khẩu</b>	<input type="password" value="....."/>
<b>Confirm Password</b>	<input type="password" value="....."/>
<b>Email</b>	<input type="text" value="trailangvocmach@gmail.com"/>
<b>Confirm Email</b>	<input type="text" value="trailangvocmach@gmail.com"/>
<b>Image Verification</b>	<div></div>

thông tin trên chú ý là email dùng là [vndprograminges@gmail.com](mailto:vndprograminges@gmail.com).

do bởi email trailang đã được sử dụng.

Tuy nhiên pass vẫn theo nguyên tắc! Characterstring171117

>>>

**#Web for technologies updating!**

>>

>>>

<https://spectrum.ieee.org/>

>>>

should read frequently.

>>>

## # Useful Sentences and Idioms

#>>>

**#Bill Gate:** “Bạn không nên quyết định hai lần cho một vấn đề. Hãy dành thời gian và suy nghĩ để ra một quyết định đúng đắn ngay lần đầu tiên để bạn không phải quay lại vấn đề một cách không cần thiết. Nếu bạn quá mong muốn xem xét lại các quyết định, bạn không chỉ làm ảnh hưởng đến quy trình công việc của bạn mà còn làm ảnh hưởng đến những tâm thế, động lực mà bạn đã có được của lần ra quyết định đầu tiên.”

# Trích từ sách: Yes or No Những quyết định thay đổi cuộc sống.

>>>

## # Abbreviation

>>>

**APT** (apt in ubuntu): **A**dvanced **P**ackaging **T**ool.  
**ATE**: **A**utomated **T**est **E**quipment.  
**ASCII**: **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange.  
**BJTs**: **B**ipolar **J**unction **T**ransistors.  
**CMP**: **C**hip **M**ultiprocessors.  
**CMSDK**: **C**ortex-**M** **S**ystem **D**esign **K**it.  
**CV**: **C**urriculum **V**itae.  
**DUT**: **D**evice **U**nder **T**est  
**DVOPD**: **D**ual **V**ideo **O**bjective **P**lane **D**ecoder **A**pplication.  
**DyAdT**: **D**ynamic **A**daptive **S**witching **A**lgorithm.  
**EUT**: **E**quipment **U**nder **T**est.  
**ESL**: **E**lectronic **S**ystem **L**evel.  
**FETs**: **F**ield-**E**ffect **T**ransistors.  
**GALS**: **G**lobally **A**synchronous **L**ocally **S**ynchronous **S**ystems.  
**MOSFET**: **M**etal **O**xide **S**emiconductor **F**ET.  
**MSD & LSD**: **M**ost **S**ignificant **D**igit and **L**east **S**ignificant **D**igit.  
**NoC**: **N**etwork-**on**-**C**hip.  
**TDDb**: **T**ime **D**ependent **D**ielectric **B**reakdown.  
**UUT**: **U**nit **U**nder **T**est (**DUT**, **EUT**).

#>>>

>>>

## **# Các thuật ngữ nên biết (terminology)**

>>>

Link: <http://vlsibyjim.blogspot.com/2015/03/basic-terminology-2.html#more>

>>>

## # 3R techniques for Reference.

>>>abc.>>>

>>>

**#3R**

>>>

!# **3R** = **R**eview + **R**evise + **R**eport

>>>

>>>

>>>insert reference here!